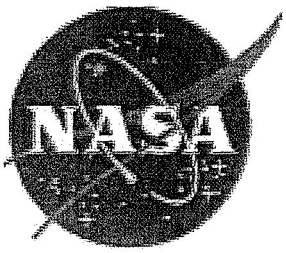




# *Incubator Display*
# *Software Cost Reduction Toolset*
# *Software Requirements Specification*

*ITC-RI-05-1001*

*October 2005*

# Table of Contents

## LIST OF TABLES

**Table** **Page**



## LIST OF FIGURES

**Figure** **Page**

# 1. INTRODUCTION

The *Incubator Display Software Requirements Specification* was initially developed by Intrinsyx Technologies Corporation (Intrinsyx) under subcontract to Lockheed Martin, Contract Number NAS2-02090, for the National Aeronautics and Space Administration (NASA) Ames Research Center (ARC) Space Station Biological Research Project (SSBRP). The second revision was approved and released in December 2003.

This *Software Cost Reduction Toolset Incubator Display Software Requirements Specification*, (SCR SRS), was modified to include efforts done under the Fiscal Year (FY) 2005 Research Infusion Project, *Application of Software Cost Reduction Tools and Methods to On-Orbit Crew Displays*, a joint collaboration done by Intrinsyx and the Naval Research Laboratory (NRL), Washington D.C. The Research Infusion Project was funded by the NASA Office of Safety and Mission Assurance (OSMA) Software Applications Research Project (SARP).

## 1.1 SCOPE

The inputs from the December 2003 SRS resulted in requirements verification, detailed in the *Application of Software Cost Reduction Tools and Methods to On-Orbit Crew Displays Final Report.* The derived parameters, listed below, are included in Appendix B.

- Type Dictionary
- Mode Class Dictionary
- Constant Dictionary
- Monitored Variable Dictionary
- Controlled Variable Dictionary
- Assumption Dictionary
- Event Function
- Mode Transition Function

The requirements were validated with the SCR Toolset. The validation methodology was scenario development from the SRS Use Cases, included in Appendix A of this document. The scenarios include:

- Actions associated with changing chamber science temperatures in a manner consistent with the Use Case defined in the initial Incubator Display SRS
- Actions associated with changing chamber fan speed in a manner consistent with the Use Case defined in the initial Incubator Display SRS
- Starting, stopping, suspending, and resuming an experiment in a manner consistent with the Use Cases defined in the initial Incubator Display SRS
- Examples of using environmental assumptions requiring that commands are acknowledged by the Incubator no later than Confirm Time after receipt

- Examples of assertions (properties) that can be proven
- Correct type checking

The SCR Toolset Application Model is included in Appendix B. The Simulation Interface, which is the Graphical User Interface (GUI), is included in Appendix C.

## 1.2 IDENTIFICATION

The Incubator Display is a User Payload Application (UPA) used to control an Incubator subrack payload for the SSBRP. The Incubator Display functions on-orbit as part of the subrack payload laptop, on the ground as part of the Communication and Data System (CDS) ground control system, and also as part of the crew training environment.

## 1.3 SYSTEM OVERVIEW

The Incubator UPA is a Java application developed for the SSBRP, and is a component of the biological laboratory developed by the SSBRP for use on-orbit on the International Space Station (ISS). The Incubator UPA resides on a laptop computer running the Windows NT (or equivalent) operating system. The laptop is a component of the SSBRP host system, and connects to it using Ethernet. The host system contains as one of its payloads an Incubator unit. The Incubator is a habitat drawer that may be inserted into a host system in order to provide a temperature-controlled environment for an experiment. The Incubator UPA is used by the crew to control the Incubator unit, and to monitor its operation and the operation of the experiment contained in the Incubator experiment chamber.

## 1.4 DOCUMENT OVERVIEW

The remainder of this document is organized as follows:

- Section 2 lists the requirements.
- Section 3 describes qualification provisions.
- Section 4 delineates requirements traceability.
- Section 5 contains a list of acronyms.
- Appendix A contains Use Cases.
- Appendix B contains the SCR Toolset Application Model.
- Appendix C contains the SCR Toolset Simulator Interface.

# 2. REQUIREMENTS

## 2.1 REQUIRED STATES AND MODES

### 2.1.1 Incubator UPA Modes Overview

The Incubator UPA operates in several different operational modes, including:

a. Flight Mode

b. Ground Mode

c. Training Mode

In each of these modes the Incubator UPA GUI operates in substantially the same way with only some minor differences. The details of these differences are described in the following subsections.

#### 2.1.1.1 Flight Mode

Flight Mode describes using the Incubator UPA as part of a host laptop while on orbit at the International Space Station (ISS). The Incubator UPA provides a set of displays allowing the astronaut to command the Incubator Payload and to display resulting Health & Status and Telemetry values. In Flight Mode the Incubator UPA communicates with the host laptop interface software using a Named Pipe protocol to receive telemetry and send commands. The rest of this document primarily describes the Flight Mode, with capabilities for other modes noted as needed.

#### 2.1.1.2 Ground Mode

Ground Mode describes using the Incubator UPA as part of the CDS of the Telescience Support Center (TSC). In Ground Mode the Incubator UPA integrates with the CDS application to receive telemetry and send commands. Ground Mode provides the same commanding and display capabilities as Flight Mode and some additional capabilities for accessing historical data, creating command sequences, and other functions.

#### 2.1.1.3 Training Mode

Training Mode allows the Incubator UPA to be used in training astronauts. In Training Mode the Incubator UPA integrates with an Incubator simulator to be able to simulate sending commands and to receive simulated telemetry. Training for CDS operators is TBD.

## 2.2 SOFTWARE ITEM CAPABILITY REQUIREMENTS

The Incubator UPA must have the general capability of controlling and monitoring the Incubator. Requirements for capabilities related to the software interface are listed in section 3.3.

### 2.2.1 Data Refresh

inc_dr_a. The display refresh rate for crew and science data shall be configurable between 3 and 60 seconds with a 1 second resolution.

inc_dr_b. Deleted.

inc_dr_c. Manual refresh capability shall be provided to allow the crew to request a refresh of the data display on demand.

inc_dr_d. Crew shall receive visual confirmation of the success or failure of a command within five seconds.

## 2.3 SOFTWARE ITEM EXTERNAL INTERFACE REQUIREMENTS

### 2.3.1 Interface Identification and Diagrams

Figure 2.3.1-1 shows the interface path between the Incubator UPA and the Incubator in Flight Mode. Messages exchanged between the two pass through two intermediaries: the host system and the interface software on the host laptop. A limited number of messages are passed between the Incubator UPA and host system via the host laptop interface software.

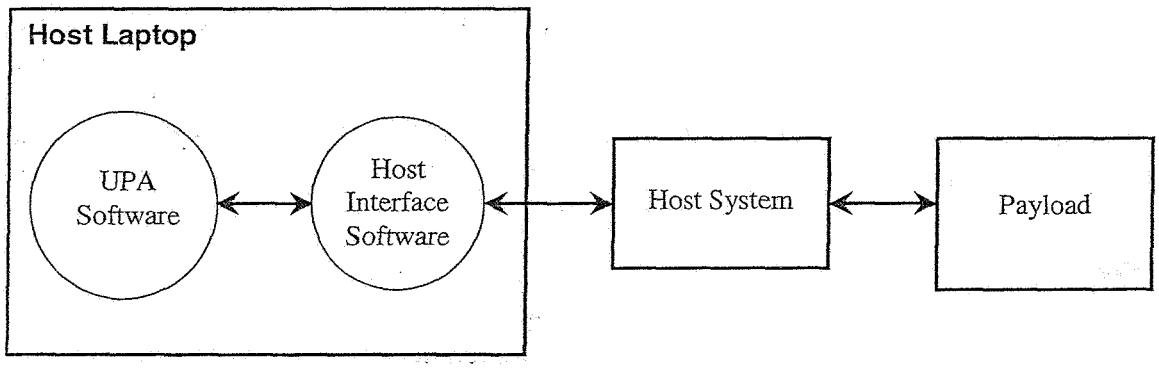



**Figure 2.3.1-1 Incubator UPA Interface – Flight Mode**

Figure 2.3.1-2 shows the interface path between the Incubator UPA and the Incubator in Ground Mode.

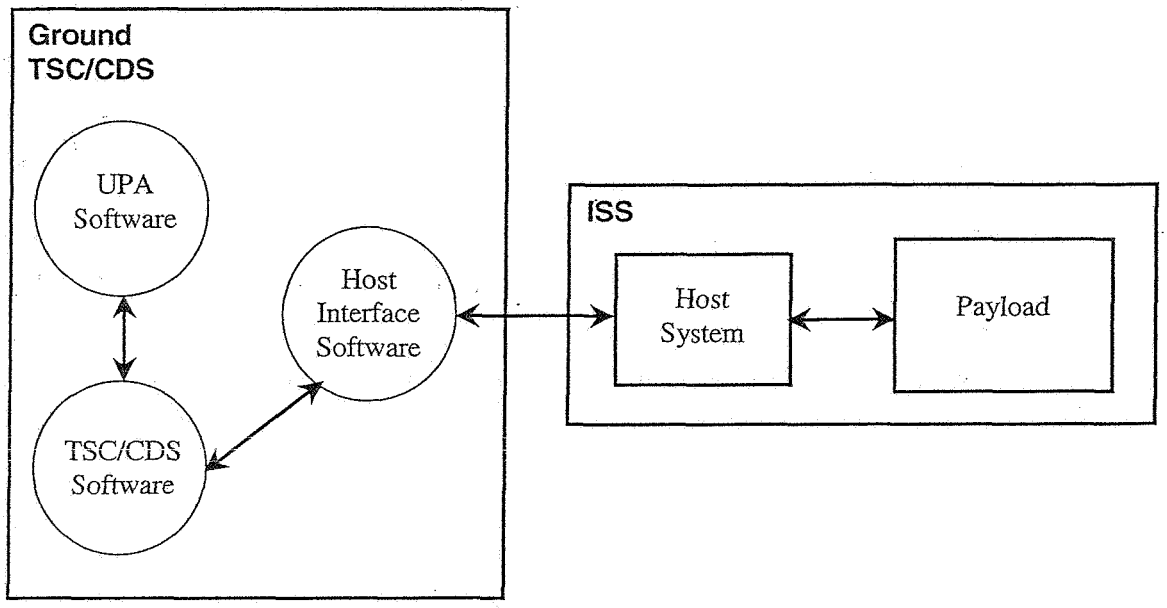



**Figure 2.3.1-2 Incubator UPA Interface – Ground Mode**

Figure 2.3.1-3 shows the interface path between the Incubator UPA and the Incubator Simulator in Training Mode.

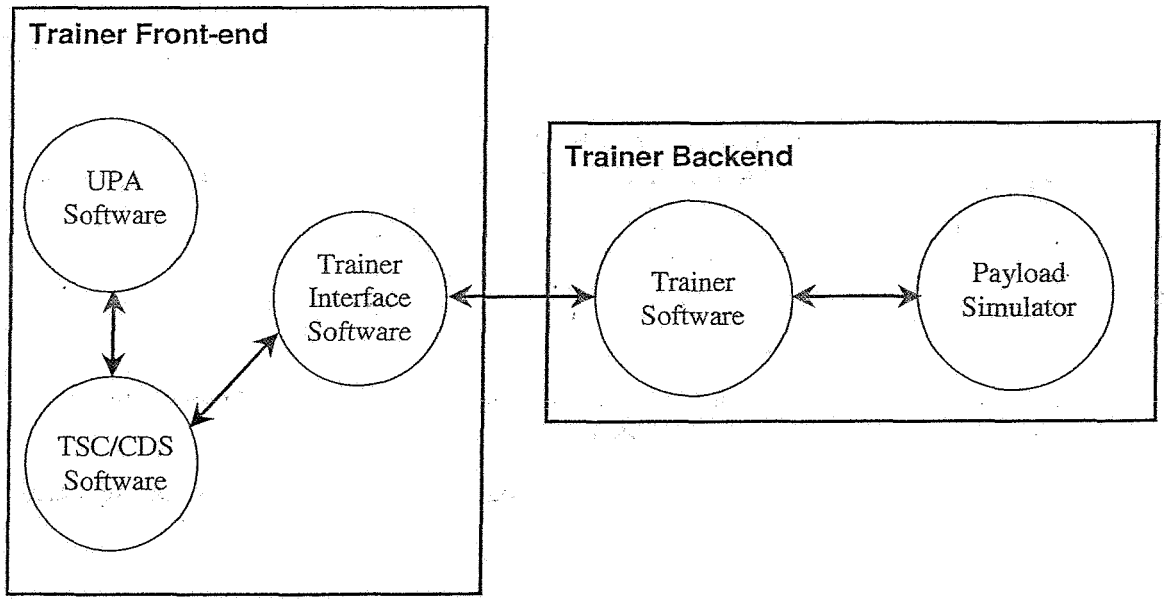



**Figure 2.3.1-3 Incubator UPA Interface – Training Mode**

### 2.3.2 Software Interface with the Incubator

The software interface with the Incubator is defined in the Incubator Interface Control Document (IICD). The following requirements derive from the IICD.

#### 2.3.2.1 Incubator Commanding

inc_ic_a. The Incubator UPA shall be capable of managing execution of command sequences on the Incubator. This includes starting execution, canceling execution, suspending execution, resuming execution, and deleting of command sequences loaded onto the Incubator.

inc_ic_b. The Incubator UPA shall be capable of commanding the operational mode of the Incubator.

inc_ic_c. The Incubator UPA shall be capable of setting the speed of the chamber fan.

inc_ic_d. The Incubator UPA shall be capable of enabling and disabling power to the chamber's two science connectors.

inc_ic_e. The Incubator UPA shall be capable of commanding the chamber reference temperature.

inc_ic_f. The Incubator UPA shall be capable of toggling the state of the chamber digital input (also called the experiment trigger).

inc_ic_g. The Incubator UPA shall be capable of setting the baud rate of the Incubator RS-422 line.

inc_ic_h. The Incubator UPA shall be capable of selecting the temperature sensors used for deriving the Incubator "Actual Temperature" value.

inc_ic_i. The Incubator UPA shall be capable of deleting and modifying time-based commands.

inc_ic_j. The Incubator UPA shall be capable of setting Incubator parameters. The Incubator defines 181 distinct parameters. In Flight Mode, the Incubator UPA shall support a subset (TBD) of the parameters. In Ground Mode, the Incubator UPA shall support all defined parameters.

inc_ic_k. In Ground Mode, the Incubator UPA shall be capable of creating Incubator command sequence commands.

inc_ic_l. In Ground Mode, the Incubator UPA shall be capable of loading Incubator command sequences into the Incubator.

inc_ic_m. In Ground Mode, the Incubator UPA shall be capable of creating Incubator command sequence files.

#### 2.3.2.2 Incubator Requests and Responses

inc_rr_a. The Incubator UPA shall be capable of requesting a telemetry data message from the Incubator and processing the response.

inc_rr_b. The Incubator UPA shall be capable of requesting an Incubator self-test and processing the response message.

inc_rr_c. The Incubator UPA shall be capable of requesting a command list from the Incubator and processing the response message.

inc_rr_d. The Incubator UPA shall be capable of requesting the file transfer of a new Incubator habitat executable from the host to the Incubator.

inc_rr_e. The Incubator UPA shall be capable of requesting the Incubator to load a command sequence file from the host.

inc_rr_f. The Incubator UPA shall be capable of requesting the values for Incubator parameters. In Flight Mode, a subset (TBD) of all the defined parameters shall be requestable. In Ground Mode, all defined parameters shall be requestable.

#### 2.3.2.3 Telemetry Reporting

inc_tr_a. The Incubator UPA shall report all crew-relevant science and engineering parameters in the Incubator telemetry message which includes:

| *Requirement* | *Sub-Requirement* |
|---|---|
| Alerts | Caution and warning word |
| | Temperature out-of-limits value |
| | Avionics fan stuck condition |
| | Chamber fan stuck condition |
| Operational Mode of the Incubator | |
| Fans | Avionics fan state – on or off |
| | Chamber fan speed – one of five levels |
| State of the Door | Open or closed |
| Electrical System | Power bus used – host or battery |
| | Total current use by fan, heater, and cooler |
| | Battery voltage |
| | Host bus voltage |
| Result of Last Self-Test | |
| Six Chamber Temperatures | Heat exchanger |
| | Coolant-in |
| | Coolant out |
| | Battery pack |
| | Avionics in |
| | Avionics out |
| Experiment Status | On/off state for each of two science connector |
| | Current draw for each of two science connector |

| | |
|---|---|
| | Two humidity sensors |
| | Experiment start and stop time |

| Requirement | Sub-Requirement |
|---|---|
| Experiment Status - continued | Four optional chamber analog monitors |
| | Five optional science temperatures |
| | State of chamber digital input (aka experiment trigger) |
| Temperature Control | Reference (commanded) temperature |
| | Actual (measured and average) temperature |
| | Temperature sensors selected for averaging |
| | Inlet cooling stare (low or moderate) |

### 2.3.3 Software Interface with the Host System

The interface between a host system and a UPA is defined in the Intra-Facility Interface Specification (IFIS) and IFIS Appendix A.

#### 2.3.3.1 Named Pipe Communication

inc_np_a. In Flight Mode, the Incubator UPA shall implement a named pipe connection to the host laptop software, as defined in IFIS section 3.3.4.1.3.

#### 2.3.3.2 Rack Subsystem Health and Status

inc_hs_a The Incubator UPA shall report the state of the payload keep-alive discrete that is transmitted in the Rack Subsystem Health and Status message from the host.

inc_hs_b The Incubator UPA shall report the state of the payload fire discrete that is transmitted in the Rack Subsystem Health and Status message from the host.

### 2.3.4 Communications and Data Systems Requirements

inc_cds_a The Incubator UPA shall be able to operate within the Flight, CDS, and Trainer systems.

inc_cds_b The CDS copy of the Incubator UPA shall not interfere with the processing and functionality of the CDS.

## 2.4 SOFTWARE ITEM INTERNAL INTERFACE REQUIREMENTS

There are no internal interface requirements for the Display.

## 2.5 SOFTWARE ITEM INTERNAL DATA REQUIREMENTS

Decisions about internal data are left to the implementation.

## 2.6 ADAPTATION REQUIREMENTS

No adaptation requirements have been identified for the Display.

## 2.7 SAFETY REQUIREMENTS

The Incubator UPA will comply with all NASA directives and standards for safety.

## 2.8 SECURITY AND PRIVACY REQUIREMENTS

The Incubator UPA will comply with all NASA directives and standards for security and privacy.

## 2.9 SOFTWARE ITEM ENVIRONMENTAL REQUIREMENTS

Not applicable.

## 2.10 COMPUTER RESOURCE REQUIREMENTS

The Incubator UPA is written in Java and runs on the host system laptop computer. Hardware, operating system, and Java version are subject to change as improved technology is adopted. This section specifies only the lowest acceptable versions and does not preclude use of future improved versions.

### 2.10.1 Computer Hardware Requirements

inc_hw_a The Incubator UPA shall be capable of running on the laptop computer used as the Host System Laptop, an IBM ThinkPad 760XD with:

- 166 MHz processor
- 64 megabytes of RAM memory
- 1024 x 768 screen size

### 2.10.2 Computer Hardware Resource Utilization Requirements

Requirements for maximum allowable use of processor capacity, memory capacity, input/ output device capacity, auxiliary storage device capacity, and communications/ network equipment capacities are TBD.

### 2.10.3 Computer Software Requirements

inc_sw_a The Incubator UPA shall be compatible with the Windows NT 4.0, SP 4 operating system.

inc_sw_b The Incubator UPA shall be capable of running under Java Runtime Environment (JRE) version 1.4.2 or later. [TBR]

### 2.10.4 Computer Communications Requirements

Not applicable.

## 2.11 SOFTWARE QUALITY FACTORS

The Incubator Display software shall be compliant with all applicable NASA and/or best practices standards and guidelines for Quality Assurance (QA), including, but not limited to, the quantitative factors listed below.

a. The software will have the ability to perform all required functions, as per this SRS.

b. The software will be capable of performing with correct, consistent results, as measured in a Formal Qualification Test (FQT).

c. The software will be written in such a manner that it is easily maintained, as demonstrated in a Software Code Review.

d. The software will be capable of portability across different platforms.

e. The software will be robust, as demonstrated in an FQT.

## 2.12 DESIGN AND IMPLEMENTATION CONSTRAINTS

### 2.12.1 Nomenclature

Nomenclature, acronyms, abbreviations, and engineering units in the Incubator UPA shall be consistent with the rules and listings in the Operations Nomenclature Document, SSP 50254.

### 2.12.2 Graphical User Interface Standards

The Incubator UPA GUI shall be consistent with the rules in the Display and Graphics Commonality Standard (DGCS), SSP 50313.

### 2.12.3 PDRT Review

The Incubator UPA shall pass a review by the Payload Display Review Team (PDRT).

### 2.12.4 Other Design Constraints

a. As warranted by requirements for portability, the software will be written such that classes and methods are reusable in subsequent habitat displays.

b. As warranted by requirements for portability, classes and methods from existing habitat displays will be reused.

# 3. QUALIFICATION PROVISIONS

Qualification provisions include the following:

- **Demonstration:** Operation of the Software Item, or a part of the Software Item, that relies on observable functional operation not requiring the use of instrumentation, special test equipment, or subsequent analysis.
- **Test:** Operation of the Software Item, or a part of the Software Item, using instrumentation or other special test equipment to collect data for later analysis.
- **Analysis:** Processing of accumulated data obtained from other qualification methods, for example, reduction, interpretation, or extrapolation of test results.
- **Inspection:** Visual examination of Software Item code, and documentation.
- **Special qualification methods:** Any special qualification methods for the Software Item, such as special tools, techniques, procedures, facilities, and acceptance limits.

Table 4.1 illustrates the qualification methods to be used for each of the major Display requirements. The test will consist of an FQT, to be witnessed by the Government, or a representative designated by the Government.

**Table 3-1 Qualification Methods**

| *Requirement* | *Summary* | *Method* |
|---|---|---|
| 2.2.1 | Data Refresh | Test |
| 2.3.2.1 | Incubator Commanding | Test |
| 2.3.2.2 | Incubator Requests/Responses | Test |
| 3.3.2.3 | Telemetry Reporting | Test |
| 2.3.3.1 | Named Pipe Communication | Test |
| 2.3.3.2 | Rack Subsystem Health and Status | Test |
| 2.3.4 | CDS Requirements | Test |
| 2.4.1 | Nomenclature | Inspection |
| 2.4.2 | GUI Standards | Inspection |
| 2.4.3 | PDRT Review | Inspection |
| 2.10.1 | Computer Hardware Requirements | Demonstration |
| 2.10.3 | Computer Software Requirements | Demonstration |
| 2.11 | Quality Factors | Inspection |
| 2.12 | Design and Implementation Constraints | Inspection |

| *Incubator Display Requirement ID* | *Use Case Number* |
|---|---|
| inc_tr_a | 1 |
| inc_np_a | 2 |
| inc_ic_f | 3 |
| inc_ic_h | 4 |
| inc_id_d | 5 |
| inc_ic_c | 6 |
| inc_ic_f | 7 |
| inc_ic_b | 8 |
| inc_tr_a | 9 |
| inc_hs_a, inc_hs_b, inc_tr_a | 10 |
| inc_rr_b | 11 |
| inc_ic_g | 12 |
| inc_dr_a, inc_dr_b, inc_dr_c | 13 |
| inc_rr_e | 14 |
| inc_ic_i | 15 |
| inc_rr_c | 16 |
| inc_ic_a, inc_tr_a | 17 |
| inc_ic_a, inc_tr_a | 18 |
| inc_ic_a, inc_tr_a | 19 |
| inc_ic_a, inc_tr_a | 20 |
| inc_ic_a, inc_tr_a | 21 |
| inc_cds_a, inc_cds_b | 22 |

# 4. NOTES

## 4.1 ABBREVIATIONS AND ACRONYMS

| | |
|---|---|
| ARC | Ames Research Center |
| BRP | Biological Research Project |
| CCB | Change Control Board |
| CDS | Communications and Data Systems |
| DGCS | Display and Graphics Commonality Standard |
| FQT | Formal Qualification Test |
| GUI | Graphical User Interface |
| ID | Identification |
| IFIS | Intra-Facility Interface Specification |
| IICD | Incubator Interface Control Document |
| ISS | International Space Station |
| JRE | Java Runtime Environment |
| NASA | National Aeronautics and Space Administration |
| OSMA | Office of Safety and Mission Assurance |
| PDRT | Payload Display Review Team |
| QA | Quality Assurance |
| SARP | Software Applications Research Project |
| SCR | Software Cost Reduction |
| SRD | Software Requirements Document |
| SRS | Software Requirements Specification |
| SSBRP | Space Station Biological Research Project |
| TBD | To Be Determined |
| TBR | To Be Revised |
| UPA | User Payload Application |

# APPENDIX A: FQT VERIFICATION USE CASES

| Use Case Number 1 | |
|---|---|
| | Monitor Chamber Status |
| **Goal in Context** | To view temperatures, relative humidity, time, alerts, laptop communication status, chamber door status, and power mode of the incubator and the experiment |
| **Scope** | Incubator Display and Incubator |
| **Level** | Summary |
| **Preconditions** | ➢ Incubator Display is properly connected to Incubator<br>➢ Incubator is functioning properly<br>➢ Incubator Display is receiving data |
| **Success End Condition** | The Incubator data can be viewed |
| **Failed End Condition** | Not able to view the data |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display has been started |
| **Main Success Scenario** | 1. Incubator hardware related information shows that Incubator Display is properly connected to the Incubator and the Incubator is functioning as expected<br>2. The Experiment window is opened and all the related information is shown |
| **Extensions** | 1a. Incubator Display shows that communications link to the Incubator is not active<br>1a1. User checks the communications link status<br>1a2. User fixes any possible communications problems with the Incubator and repeats this Use Case<br>2a. There is no data shown in the Experiment sub-window<br>2a1. User checks the Incubator and communications links, fixes any possible problems and repeats this Use Case |
| **Related Information** | ➢ Configure Incubator Display<br>➢ Adjust Chamber Temperature Use Case<br>➢ Select Temperature Sensors for Averaging Use Case<br>➢ Switch Science Power Connectors Use Case<br>➢ Change Chamber Fan Speed Use Case<br>➢ Activate/Deactivate Experiment Trigger Use Case<br>➢ Set Power Mode Use Case<br>➢ View Incubator System Information Use Case<br>➢ Monitor Alert Use Case<br>➢ Run Self-Test Use Case<br>➢ Set Baud Rate Use Case |

| | |
|---|---|
| | ➢ Set Data Refresh Rate Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 2 | |
|---|---|
| **Use Case** | Configure Incubator Display |
| **Goal in Context** | To configure the Incubator Display to match the current rack hardware configuration in order to display the temperatures, relative humidity, time, alerts, laptop communication status, chamber door status, and power mode of the Incubator and the experiment it contains |
| **Scope** | Incubator Display and Incubator |
| **Level** | Summary |
| **Preconditions** | ➢ Incubator Display is properly connected to Incubator<br>➢ Incubator is functioning properly<br>➢ Incubator Display is receiving data |
| **Success End Condition** | The Incubator Display has been configured to match a specific Incubator and is displaying data from the experiment it contains |
| **Failed End Condition** | Not able to connect to the Incubator |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display has been started |
| **Main Success Scenario** | 1. Incubator hardware related information shows that Incubator Display is properly connected to the Incubator and the Incubator is functioning as expected<br>2. The Experiment window is opened and related information from the selected Incubator is shown |
| **Extensions** | 1a. Incubator Display shows that communications link to the Incubator is not active<br>1a1. User uses a screen widget on Incubator Display to check the communications link status<br>1a2. User fixes the communications problem with the Incubator and repeats this Use Case<br>2a. There is no data shown in the Experiment window<br>2a1. User ensures that the Incubator and communications link work properly and repeats this Use Case |
| **Related Information** | ➢ Monitor Chamber Status Use Case<br>➢ Adjust Chamber Temperature Use Case<br>➢ Select Temperature Sensors for Averaging Use Case<br>➢ Switch Science Power Connectors Use Case<br>➢ Change Chamber Fan Speed Use Case<br>➢ Activate/Deactivate Experiment Trigger Use Case<br>➢ Set Power Mode Use Case |

| | |
|---|---|
| | ➢ View Incubator System Information Use Case<br>➢ Monitor Alert Use Case<br>➢ Run Self-Test Use Case<br>➢ Set Baud Rate Use Case<br>➢ Set Data Refresh Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 3 | |
|---|---|
| **Use Case** | Adjust Chamber Temperature |
| **Goal in Context** | To adjust Chamber Temperature |
| **Scope** | Incubator Display and Incubator |
| **Level** | |
| **Preconditions** | ➢ Incubator Display is properly connected to Incubator<br>➢ Incubator is functioning properly<br>➢ Incubator Display is receiving data |
| **Success End Condition** | ➢ The user is able to set temperature to a desired value<br>➢ The desired value is reflected in the Display |
| **Failed End Condition** | Not able to set the temperature |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display has been started |
| **Main Success Scenario** | 1. User opens the temperature control window<br>2. User sets the chamber temperature to a new value<br>3. User confirms the setting of the new value<br>4. User observes that the chamber temperature reaches the newly set value |
| **Extensions** | 4a. Chamber temperature value remains unchanged after the expected lag time for it to change<br>4a1. User checks the Incubator to make sure it is working properly<br>4a2. User follows Use Case Adjust Temperature Sensors to select the right combination of sensors to make sure the correct temperature values will be shown |
| **Related Information** | ➢ Monitor Chamber Status Use Case<br>➢ Configure Incubator Display Use Case<br>➢ Select Temperature Sensors for Averaging Use Case<br>➢ Switch Science Power Connectors Use Case<br>➢ Change Chamber Fan Speed Use Case<br>➢ Activate/Deactivate Experiment Trigger Use Case<br>➢ Set Power Mode Use Case<br>➢ View Incubator System Information Use Case<br>➢ Monitor Alert Use Case<br>➢ Run Self-Test Use Case<br>➢ Set Baud Rate Use Case<br>➢ Set Data Refresh Rate Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |

| | |
|---|---|
| **Frequency** | |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 4 | |
|---|---|
| Use Case | Select Temperature Sensors for Averaging |
| **Goal in Context** | To select a specific combination of temperature sensors for determining the averaged chamber temperature they represent |
| **Scope** | Incubator Display and Incubator |
| **Level** | |
| **Preconditions** | ➢ Incubator Display is properly connected to Incubator<br>➢ Incubator is functioning properly<br>➢ Incubator Display is receiving data |
| **Success End Condition** | The user is able to select a set of temperature sensors to monitor the chamber temperature |
| **Failed End Condition** | Not able to observe the expected temperature readings |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display has been started |
| **Main Success Scenario** | 1. User opens the temperature sensor selection window<br>2. User selects a new set of sensors<br>3. User confirms the selection<br>4. User observes that the correct sensors have been set |
| **Extensions** | 4a. The Incubator UPA cannot confirm the selection of the sensors<br>4a1. User checks the Incubator to make sure that the selected sensors are working properly<br>4a2. User follows Monitor Chamber Status Use Case to find out if the problem has been fixed and repeats this Use Case<br>5a. The temperature readings do not reflect the expected values for the sensor combination selected<br>5a1. User follows steps 4a1 and 4a2 |
| **Related Information** | ➢ Monitor Chamber Status Use Case<br>➢ Configure Incubator Display Use Case<br>➢ Adjust Chamber Temperature Use Case<br>➢ Switch Science Power Connectors Use Case<br>➢ Change Chamber Fan Speed Use Case<br>➢ Activate/Deactivate Experiment Trigger Use Case<br>➢ Set Power Mode Use Case<br>➢ View Incubator System Information Use Case<br>➢ Monitor Alert Use Case<br>➢ Run Self-Test Use Case<br>➢ Set Baud Rate Use Case |

| | |
|---|---|
| | ➢ Set Data Refresh Rate Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 5 | |
|---|---|
| **Use Case** | Switch Science Power Connector(s) |
| **Goal in Context** | *To turn the science power connection(s) on or off* |
| **Scope** | Incubator Display and Incubator |
| **Level** | |
| **Preconditions** | ➢ Incubator Display is properly connected to Incubator<br>➢ Incubator is functioning properly<br>➢ Incubator Display is receiving data |
| **Success End Condition** | The user is able to turn on or off the connector(s) |
| **Failed End Condition** | Not able to turn the connector(s) on or off |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display has been started |
| **Main Success Scenario** | 1. User sets the Science Power Connector on the Incubator front panel<br>2. User opens the Science Power Connector window<br>3. User turns on or off the connector(s) to match the setting on the Incubator front panel<br>4. User confirms the action<br>5. User observes that the connector(s) has been turned on or off and the Incubator Display reflects the change |
| **Extensions** | 5a. User finds out the connector state has not changed<br>4a1. User ensures connector control works properly<br>4a2. User repeats this Use Case if necessary |
| **Related Information** | ➢ Monitor Chamber Status Use Case<br>➢ Configure Incubator Display Use Case<br>➢ Adjust Chamber Temperature Use Case<br>➢ Select Temperature Sensors for Averaging Use Case<br>➢ Change Chamber Fan Speed Use Case<br>➢ Activate/Deactivate Experiment Trigger Use Case<br>➢ Set Power Mode Use Case<br>➢ View Incubator System Information Use Case<br>➢ Monitor Alert Use Case<br>➢ Run Self-Test Use Case<br>➢ Set Baud Rate Use Case<br>➢ Set Data Refresh Rate Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | |
| ***Channel to Actors*** | Interactive |
| **Due Date** | |

| | |
|---|---|
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 6 | |
|---|---|
| **Use Case** | Change Chamber Fan Speed |
| **Goal in Context** | To set the Chamber Fan to a desired speed |
| **Scope** | Incubator Display and Incubator |
| **Level** | |
| **Preconditions** | ➢ Incubator Display is properly connected to Incubator<br>➢ Incubator is functioning properly<br>➢ Incubator Display is receiving data |
| **Success End Condition** | Chamber fan is set to a desired speed |
| **Failed End Condition** | Not able to turn the fan to a desired speed |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display has been started |
| **Main Success Scenario** | 1. User opens the Chamber Fan Speed window<br>2. User sets the fan to a desired speed<br>3. User confirms the action<br>4. User observes that the fan is operating at the new speed |
| **Extensions** | 4a. User finds out that the fan speed has not changed<br>4a1. User makes sure the fan is working properly<br>4a2. User repeats this Use Case if necessary |
| **Related Information** | ➢ Monitor Chamber Status Use Case<br>➢ Configure Incubator Display Use Case<br>➢ Adjust Chamber Temperature Use Case<br>➢ Select Temperature Sensors for Averaging Use Case<br>➢ Switch Science Power Connector(s) Use Case<br>➢ Activate/Deactivate Experiment Trigger Use Case<br>➢ Set Power Mode Use Case<br>➢ View Incubator System Information Use Case<br>➢ Monitor Alert Use Case<br>➢ Run Self-Test Use Case<br>➢ Set Baud Rate Use Case<br>➢ Set Data Refresh Rate Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 7 | |
|---|---|
| **Use Case** | Activate/Deactivate Experiment Trigger |
| **Goal in Context** | To activate/deactivate digital control line to experiment |
| **Scope** | Incubator Display and Incubator |
| **Level** | |
| **Preconditions** | ➢ Incubator Display is properly connected to Incubator<br>➢ Incubator is functioning properly<br>➢ Incubator Display is receiving data |
| **Success End Condition** | The experiment is activated or deactivated |
| **Failed End Condition** | Not able to activate or deactivate an experiment |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display has been started |
| **Main Success Scenario** | 1. User opens the Experiment Trigger window<br>2. User activates or deactivates the experiment trigger<br>3. User confirms the action<br>4. User observes the new state of the experiment trigger |
| **Extensions** | |
| **Related Information** | ➢ Monitor Chamber Status Use Case<br>➢ Configure Incubator Display Use Case<br>➢ Adjust Chamber Temperature Use Case<br>➢ Select Temperature Sensors for Averaging Use Case<br>➢ Switch Science Power Connector(s) Use Case<br>➢ Change Chamber Fan Speed Use Case<br>➢ Set Power Mode Use Case<br>➢ View Incubator System Information Use Case<br>➢ Monitor Alert Use Case<br>➢ Run Self-Test Use Case<br>➢ Set Baud Rate Use Case<br>➢ Set Data Refresh Rate Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 8 | |
|---|---|
| Use Case | Set Power Mode |
| **Goal in Context** | To set Incubator power to be in Normal or Idle mode |
| **Scope** | Incubator Display and Incubator |
| **Level** | |
| **Preconditions** | ➢ Incubator Display is properly connected to Incubator<br>➢ Incubator is functioning properly<br>➢ Incubator Display is receiving data |
| **Success End Condition** | The power mode is changed |
| **Failed End Condition** | Not able to change the mode |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display has been started |
| **Main Success Scenario** | 1. User opens the Power Mode window<br>2. User sets the mode to be Normal or Idle<br>3. User confirms the action<br>4. User observes the new power mode |
| **Extensions** | 4a. User finds out that the power mode has not changed<br>4a1. User makes sure the power mode control function is working properly<br>4a2. User repeats this Use Case if necessary |
| **Related Information** | ➢ Monitor Chamber Status Use Case<br>➢ Configure Incubator Display Use Case<br>➢ Adjust Chamber Temperature Use Case<br>➢ Select Temperature Sensors for Averaging Use Case<br>➢ Switch Science Power Connector(s) Use Case<br>➢ Change Chamber Fan Speed Use Case<br>➢ Activate/Deactivate Experiment Trigger Use Case<br>➢ View Incubator System Information<br>➢ Monitor Alert Use Case<br>➢ Run Self-Test Use Case<br>➢ Set Baud Rate Use Case<br>➢ Set Data Refresh Rate Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 9 | |
|---|---|
| **Use Case** | View Incubator System Information |
| **Goal in Context** | To view all the system level information of Incubator |
| **Scope** | Incubator Display and Incubator |
| **Level** | Summary |
| **Preconditions** | ➢ Incubator Display is properly connected to Incubator<br>➢ Incubator is functioning properly<br>➢ Incubator Display is receiving data |
| **Success End Condition** | The system information is displayed |
| **Failed End Condition** | Not able to view the system information |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display has been started |
| **Main Success Scenario** | 1. User opens the System Information window<br>2. Incubator system information is displayed |
| **Extensions** | 2a. User finds out that system information displayed does not reflect the actual status of the Incubator<br>2a1. User makes sure all system components, such as Incubator and Incubator Display, are working properly as expected<br>2a2. User repeats this Use Case if necessary |
| **Related Information** | ➢ Monitor Chamber Status Use Case<br>➢ Configure Incubator Display Use Case<br>➢ Adjust Chamber Temperature Use Case<br>➢ Select Temperature Sensors for Averaging Use Case<br>➢ Switch Science Power Connector(s) Use Case<br>➢ Change Chamber Fan Speed Use Case<br>➢ Activate/Deactivate Experiment Trigger Use Case<br>➢ Set Power Mode Use Case<br>➢ Monitor Alert Use Case<br>➢ Run Self-Test Use Case<br>➢ Set Baud Rate Use Case<br>➢ Set Data Refresh Rate Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 10 | |
|---|---|
| **Use Case** | Monitor Alert |
| **Goal in Context** | To check the content of the alerts |
| **Scope** | Incubator Display and Incubator |
| **Level** | |
| **Preconditions** | ➢ Incubator Display is properly connected to Incubator<br>➢ Incubator is functioning properly<br>➢ Incubator Display is receiving data |
| **Success End Condition** | User is able to view the content of the alerts |
| **Failed End Condition** | Not able to view the content of the alerts |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display has been started |
| **Main Success Scenario** | 1. User notices the alert display area indicates that there are alerts<br>2. User opens the window that displays alert details<br>3. User decides whether to take actions based on the pre-defined operating procedures |
| **Extensions** | |
| **Related Information** | ➢ Monitor Chamber Status Use Case<br>➢ Configure Incubator Display Use Case<br>➢ Adjust Chamber Temperature Use Case<br>➢ Select Temperature Sensors for Averaging Use Case<br>➢ Switch Science Power Connector(s) Use Case<br>➢ Change Chamber Fan Speed Use Case<br>➢ Activate/Deactivate Experiment Trigger Use Case<br>➢ View Incubator System Information<br>➢ Set Power Mode Use Case<br>➢ Run Self-Test Use Case<br>➢ Set Baud Rate Use Case<br>➢ Set Data Refresh Rate Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 11 | |
|---|---|
| Use Case | Run Self-Test |
| Goal in Context | To make Incubator run its Self-Test program |
| Scope | Incubator Display and Incubator |
| Level | |
| Preconditions | ➢ Incubator Display is properly connected to Incubator<br>➢ Incubator is functioning properly<br>➢ Incubator Display is receiving data |
| Success End Condition | The Incubator Self-Test program is able to run |
| Failed End Condition | Not able to make the Incubator to run the Self-Test |
| Primary Actor | Incubator Display user |
| Secondary Actor | Incubator |
| Stakeholders/Interests | |
| Trigger | Incubator Display user issues the screen command to start the Self-Test |
| Main Success Scenario | 1. User opens the Incubator Self-Test window<br>2. User issues the screen command to start the Test<br>3. The Incubator UPA shows the Self-Test results |
| Extensions | 3a. Self-Test results indicate errors or failure<br>3a1. User finds out the cause(s), fixes the problem(s) and repeats this Use Case |
| Related Information | ➢ Monitor Chamber Status Use Case<br>➢ Configure Incubator Display Use Case<br>➢ Adjust Chamber Temperature Use Case<br>➢ Select Temperature Sensors for Averaging Use Case<br>➢ Switch Science Power Connector(s) Use Case<br>➢ Change Chamber Fan Speed Use Case<br>➢ Activate/Deactivate Experiment Trigger Use Case<br>➢ Set Power Mode Use Case<br>➢ View Incubator System Information<br>➢ Monitor Alert Use Case<br>➢ Set Baud Rate Use Case<br>➢ Set Data Refresh Rate Use Case |
| Priority | Top |
| Performance | <= 2 seconds |
| Frequency | |
| Channel to Actors | Interactive |
| Due Date | |
| Super-ordinates | |
| Subordinates | |

| Use Case Number 12 | |
|---|---|
| **Use Case** | Set Baud Rate |
| **Goal in Context** | To set the baud rate for the communications link between the Incubator Display and Incubator |
| **Scope** | Incubator Display and Incubator |
| **Level** | |
| **Preconditions** | The baud rate selected is supported by both Incubator Display and Incubator |
| **Success End Condition** | A new baud rate between the Incubator Display and Incubator is set |
| **Failed End Condition** | Not able to set the new baud rate |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display opens the Baud Rate Selection Window |
| **Main Success Scenario** | 1. User opens the Set Baud Rate window<br>2. User selects a new baud rate<br>3. User confirms his/her selection<br>4. Incubator Display confirms effective new baud rate |
| **Extensions** | 4a. The baud rate selected is not supported by Incubator<br>4a1. User finds out the baud rates supported by the Incubator and repeats this Use Case |
| **Related Information** | ➢ Monitor Chamber Status Use Case<br>➢ Configure Incubator Display Use Case<br>➢ Adjust Chamber Temperature Use Case<br>➢ Select Temperature Sensors for Averaging Use Case<br>➢ Switch Science Power Connector(s) Use Case<br>➢ Change Chamber Fan Speed Use Case<br>➢ Activate/Deactivate Experiment Trigger Use Case<br>➢ Set Power Mode Use Case<br>➢ View Incubator System Information<br>➢ Monitor Alert Use Case<br>➢ Run Self-Test Use Case<br>➢ Set Data Refresh Rate Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 13 | |
|---|---|
| Use Case | Set Data Refresh Rate |
| **Goal in Context** | To set the rate that Incubator Display refreshes its data display |
| **Scope** | Incubator Display and Incubator |
| **Level** | |
| **Preconditions** | ➢ Incubator Display is properly connected to Incubator<br>➢ Incubator is functioning properly<br>➢ Incubator Display is receiving data<br>➢ The desired rate exists |
| **Success End Condition** | The desired rate has been set |
| **Failed End Condition** | Not able to set the new rate |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Set Data Refresh Rate Manager Window has been opened |
| **Main Success Scenario** | 1. User opens the Set Date Refresh Rate Manager window<br>2. User selects a new rate<br>3. User confirms his/her selection<br>4. Incubator Display refreshes data with new rate |
| **Extensions** | |
| **Related Information** | ➢ Monitor Chamber Status Use Case<br>➢ Configure Incubator Display Use Case<br>➢ Adjust Chamber Temperature Use Case<br>➢ Select Temperature Sensors for Averaging Use Case<br>➢ Switch Science Power Connector(s) Use Case<br>➢ Change Chamber Fan Speed Use Case<br>➢ Activate/Deactivate Experiment Trigger Use Case<br>➢ Set Power Mode Use Case<br>➢ View Incubator System Information<br>➢ Monitor Alert Use Case<br>➢ Run Self-Test Use Case<br>➢ Set Baud Rate Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 14 | |
|---|---|
| **Use Case** | View Time-Based Command Details |
| **Goal in Context** | To view the details of Time-Based Commands from the Time-Based Command list |
| **Scope** | Incubator Display and Incubator |
| **Level** | |
| **Preconditions** | The Command exists in the Incubator |
| **Success End Condition** | The Command details are displayed |
| **Failed End Condition** | Not able to display the Command details |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Time-Based Command Manager Window has been opened |
| **Main Success Scenario** | 1. User opens Time-Based Command Manager window<br>2. User selects the Command to be viewed<br>3. User chooses to view the details of the Command<br>4. The details of the Command is displayed |
| **Extensions** | |
| **Related Information** | Delete Time-Based Command Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 15 | |
|---|---|
| Use Case | Delete Time-Based Command |
| Goal in Context | To delete a Time-Based Command from the Time-Based Command list |
| Scope | Incubator Display and Incubator |
| Level | |
| Preconditions | ➢ Command to be deleted exists in the Incubator<br>➢ User knows which command to delete |
| Success End Condition | The command selected for deletion has been deleted |
| Failed End Condition | Not able to delete the command |
| Primary Actor | Incubator Display user |
| Secondary Actor | Incubator |
| Stakeholders/Interests | |
| Trigger | Time-Based Command Manager Window has been opened |
| Main Success Scenario | 1. User opens Time-Based Command Manager window<br>2. User selects the command to be deleted<br>3. User confirms his/her intention<br>4. Incubator deletes the command from its record<br>5. Command list shows selected command was deleted |
| Extensions | 2a. User cannot find command that he/she is looking for<br>3a1. User makes sure that he/she is looking for the right command and repeats this Use Case |
| Related Information | View Time-Based Command Details Use Case |
| Priority | Top |
| Performance | <= 2 seconds |
| Frequency | |
| Channel to Actors | Interactive |
| Due Date | |
| Super-ordinates | |
| Subordinates | |

| Use Case Number 16 | |
|---|---|
| Use Case | View Command Sequence Details |
| **Goal in Context** | To view the details of a command sequence |
| **Scope** | Incubator Display and Incubator |
| **Level** | |
| **Preconditions** | The command sequence and its details exist in the command sequence list |
| **Success End Condition** | The details of a command sequence is displayed |
| **Failed End Condition** | Not able to view the details of the command sequence |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display has been started |
| **Main Success Scenario** | 1. User opens the Command Sequence Manager window<br>2. User selects Command Sequence of interest<br>3. User chooses to view the details of the Sequence<br>4. The details of the Sequence is displayed |
| **Extensions** | |
| **Related Information** | ➢ Run Command Sequence Use Case<br>➢ Cancel Command Sequence Use Case<br>➢ Suspend Command Sequence Use Case<br>➢ Resume Command Sequence Use Case<br>➢ Delete Command Sequence Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | Ad hoc |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 17 | |
|---|---|
| **Use Case** | Run Command Sequence |
| **Goal in Context** | To execute a command sequence |
| **Scope** | Incubator Display and Incubator |
| **Level** | |
| **Preconditions** | ➢ Incubator Display is properly connected to Incubator<br>➢ Incubator is functioning properly<br>➢ Incubator Display is receiving data<br>➢ The command sequence exists in the command sequence list<br>➢ User knows which command sequence to run |
| **Success End Condition** | A command sequence is executed |
| **Failed End Condition** | Not able to execute the command sequence |
| **Primary Actor** | *Incubator Display user* |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display has been started |
| **Main Success Scenario** | 1. User opens the Command Sequence Manager window<br>2. Only those command sequences that can be run under the circumstances are shown to be selectable<br>3. User selects the command sequence to run<br>4. User confirms his/her intention<br>5. The response(s) from Incubator indicates that the command sequence has been executed |
| **Extensions** | 5a. Incubator is not responding<br>5a1. User makes sure the system is functioning properly and repeats this Use Case<br>5b. Incubator indicates a problem executing the sequence<br>5b1. User follows 5a1 |
| **Related Information** | ➢ View Command Sequence Details Use Case<br>➢ Cancel Command Sequence Use Case<br>➢ Suspend Command Sequence Use Case<br>➢ Resume Command Sequence Use Case<br>➢ Delete Command Sequence Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 18 | |
|---|---|
| **Use Case** | Cancel Command Sequence |
| **Goal in Context** | To cancel a command sequence that just has been executed |
| **Scope** | Incubator Display and Incubator |
| **Level** | |
| **Preconditions** | ➢ Incubator Display is properly connected to Incubator<br>➢ Incubator is functioning properly<br>➢ Incubator Display is receiving data<br>➢ User knows which command sequence to cancel |
| **Success End Condition** | The execution of a command sequence is canceled |
| **Failed End Condition** | Not able to cancel the command sequence |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display has been started |
| **Main Success Scenario** | 1. User opens the Command Sequence Manager window<br>2. Only those command sequences that can be canceled under the circumstances are shown to be selectable<br>3. User selects the command sequence to cancel<br>4. User confirms his/her intention<br>5. The response(s) from Incubator indicates that the command sequence has been canceled |
| **Extensions** | 5a. Incubator is not responding<br>5a1. User makes sure the system is functioning properly and repeats this Use Case<br>5b. Incubator indicates a problem canceling the sequence<br>5b1. User follows 5a1 |
| **Related Information** | ➢ View Command Sequence Details Use Case<br>➢ Run Command Sequence Use Case<br>➢ Suspend Command Sequence Use Case<br>➢ Resume Command Sequence Use Case<br>➢ Delete Command Sequence Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | Ad hoc |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 19 | |
|---|---|
| **Use Case** | Suspend Command Sequence |
| **Goal in Context** | To suspend a command sequence that is executing |
| **Scope** | Incubator Display and Incubator |
| **Level** | |
| **Preconditions** | ➢ Incubator Display is properly connected to Incubator<br>➢ Incubator is functioning properly<br>➢ Incubator Display is receiving data<br>➢ User knows which command sequence to suspend |
| **Success End Condition** | The execution of a command sequence is suspended |
| **Failed End Condition** | Not able to suspend the Command Sequence |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display has been started |
| **Main Success Scenario** | 1. User opens the Command Sequence window<br>2. Only those command sequences that can be suspended under the circumstances are shown to be selectable<br>3. User selects the command sequence to suspend<br>4. User confirms his/her intention<br>5. The response(s) from Incubator indicates that the command sequence has been suspended |
| **Extensions** | 5a. Incubator is not responding<br>5a1. User makes sure the system is functioning properly and repeats this Use Case<br>5b. Incubator indicates a problem suspending the sequence<br>5b1. User follows 5a1 |
| **Related Information** | ➢ View Command Sequence Details Use Case<br>➢ Run Command Sequence Use Case<br>➢ Cancel Command Sequence Use Case<br>➢ Resume Command Sequence Use Case<br>➢ Delete Command Sequence Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | Ad hoc |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 20 | |
|---|---|
| **Use Case** | Resume Command Sequence |
| **Goal in Context** | To resume a command sequence that has been suspended |
| **Scope** | Incubator Display and Incubator |
| **Level** | |
| **Preconditions** | ➢ Incubator Display is properly connected to Incubator<br>➢ Incubator is functioning properly<br>➢ Incubator Display is receiving data<br>➢ User knows which command sequence to resume |
| **Success End Condition** | Suspended execution of a command sequence is resumed |
| **Failed End Condition** | Not able to resume the suspended command sequence |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display has been started |
| **Main Success Scenario** | 1. User opens the Command Sequence Manager window<br>2. Only those command sequences that can be resumed under the circumstances are shown to be selectable<br>3. User selects the command sequence to resume<br>4. User confirms his/her intention<br>5. The response(s) from Incubator indicates that the execution of the command sequence has been resumed |
| **Extensions** | 5a. Incubator is not responding<br>5a1. User makes sure the system is functioning properly and repeats this Use Case<br>5b. Incubator indicates that it cannot resume selected sequence<br>5b1. User makes sure that he/she is resuming the right command sequence and repeats this Use Case |
| **Related Information** | ➢ View Command Sequence Details Use Case<br>➢ Run Command Sequence Use Case<br>➢ Cancel Command Sequence Use Case<br>➢ Suspend Command Sequence Use Case<br>➢ Delete Command Sequence Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | Ad hoc |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 21 | |
|---|---|
| **Use Case** | Delete Command Sequence |
| **Goal in Context** | To delete a command sequence from the command sequence list |
| **Scope** | Incubator Display and Incubator |
| **Level** | |
| **Preconditions** | ➢ The command sequence exists in the Incubator<br>➢ User knows which command sequence to delete |
| **Success End Condition** | The command sequence is deleted |
| **Failed End Condition** | Not able to delete the command sequence |
| **Primary Actor** | Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders/Interests** | |
| **Trigger** | Incubator Display has been started |
| **Main Success Scenario** | 1. User opens the Command Sequence window<br>2. Only those command sequences that can be deleted under the circumstances are shown to be selectable<br>3. User selects the command sequence to delete<br>4. User confirms his/her intention<br>5. Incubator deletes the sequence<br>6 Incubator Display confirms that the command sequence has been deleted |
| **Extensions** | |
| **Related Information** | ➢ View Command Sequence Details Use Case<br>➢ Run Command Sequence Use Case<br>➢ Cancel Command Sequence Use Case<br>➢ Suspend Command Sequence Use Case<br>➢ Resume Command Sequence Use Case |
| **Priority** | Top |
| **Performance** | <= 2 seconds |
| **Frequency** | Ad hoc |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

| Use Case Number 22 | |
|---|---|
| **Use Case** | Invoke Incubator Display in CDS |
| **Goal in Context** | To run Incubator Display and view Incubator data in CDS |
| **Scope** | Incubator Display, Incubator, and CDS |
| **Level** | |
| **Preconditions** | ➢ CDS is running<br>➢ The Incubator Display in CDS can receive the same data that the in-flight Incubator Display receives |
| **Success End Condition** | The Incubator Display is started in CDS and showing data from the in-flight Incubator |
| **Failed End Condition** | Not able to see the Incubator data |
| **Primary Actor** | CDS/Incubator Display user |
| **Secondary Actor** | Incubator |
| **Stakeholders and Interests** | |
| **Trigger** | CDS user issues screen command to start Incubator Display |
| **Main Success Scenario** | 1. User opens the CDS window where he/she can issue the screen command to start Incubator Display<br>2. Incubator Display starts within the context of CDS<br>3. Incubator Display shows the data from Incubator |
| **Extensions** | 3a. Incubator Display cannot show Incubator data<br>3a1. User ensures that the linkage between the Incubator Display and Incubator works and repeats this Use Case |
| **Related Information** | Incubator Display related Use Cases |
| **Priority** | Top |
| **Performance** | |
| **Frequency** | Ad hoc |
| **Channel to Actors** | Interactive |
| **Due Date** | |
| **Super-ordinates** | |
| **Subordinates** | |

# APPENDIX B: SCR TOOLSET APPLICATION MODEL

*Incubator Display*
*Software Cost Reduction Toolset Model*
*Version 5*
*October 2005*

Version 5 of the Incubator Display Software Cost Reduction (SCR) Toolset Model includes:

- Actions associated with changing chamber science temperature in a manner consistent with Use Case Number 3
- Actions associated with changing chamber fan speed in a manner consistent with Use Case Number 6
- Starting, stopping, suspending, and resuming an experiment (Use Cases 17,18,19,20)
- A complete scenario (i.e., event list provided to the simulator) that can be exercised either in the standard simulator, or executed more realistically a GUI
- Example of using environmental assumptions, requiring that commands are acknowledged by the Incubator no later than ConfirmTime after received
- Specification type checks

Changes to Revision 5, final version, from previous revisions include:

- Minor changes in units and types for time and temperature
- Change confirms to be more uniform by making them of CommandType
- Change of name mcExecutionStatus to mcTimedExecutionStatus
- New mcExecutionStatus keeps track of how many immediate commands have interrupted normal timed command sequence execution
- New mcExecutionStatus and mcTimedExecutionStatus together reflect the operational status of the Incubator with respect to the currently selected command sequence and any immediate commands, which may have interrupted normal timed command sequence execution
- Simplified definition of cCommand by triggering its changes with respect to the mode classes mcExecutionStatus and mcTimedExecutionStatus
- Changed event that indicates temperature has reached the newly commanded reference temperature to@T( mRefSciTemp - mSciTemp | <= TempErr ) to reflect tolerance in the amount TempErr
- If |mRefSciTemp - mSciTemp| <= TempErr for the new Reference Science Temperature, return to idle
- Made event controlling when changing reference science temperature into a \"signal\": @C(mEnableSciTemp), a boolean variable change

- Simplified table definitions that were essentially equivalent to cXXX = mXXX to non-table definitions that are then put directly into the variable dictionary

The SCR Toolset works with a defined set of dictionaries that consist of sets of variables and parameters against which the SCR Toolset is run for requirements validation. The Incubator Display dictionaries are delineated below. Results of the Toolset runs are included in the *Application of Software Cost Reduction Tools and Methods to On-Orbit Crew Displays Final Report.*

| *Type Dictionary* | | | | |
|---|---|---|---|---|
| ***Name*** | ***Base Type*** | ***Units*** | ***Legal Values*** | ***Comment*** |
| CommandType | Enumerated | | None, SetRefTemp, SetFan, StartExp, StopExp, ResumeExp | Commands to the Incubator |
| DoorType | Enumerated | | Closed, Open | Positions of the Incubator door |
| FanSpeed | Enumerated | | Off, Low, MedHigh, High | Valid chamber fan speeds |
| Temperature | Integer | Degrees Celsius | [0-100] | Chamber temperature range |

| Mode Class Dictionary | | | |
|---|---|---|---|
| ***Name*** | ***Type*** | ***Value*** | ***Comment*** |
| mcExecutionStatusConfirm Time | Timed, Immediate, Immediate2 | Timed | Modes indicating how many immediate commands are executing |
| mcFanMode | Idle, FanChangeEnabled, CommandSent, ChangingSciTemp, SciTempError | Idle | Modes associated with changing fan speed and actions done by the Incubator in response to change |
| McSciTempMode | Idle, TempChangeEnabled, CommandSent, ChangingSciTemp, SciTempError | Idle | Modes associated with changing reference science temperature and actions done by the Incubator in response to change |
| mcTimeExecutionStatus | Stopped, Running, Suspended | Stopped | Status of execution of current timed command sequence, if not interrupted by immediate command |

| Constant Dictionary | | | |
|---|---|---|---|
| ***Name*** | ***Type*** | ***Value*** | ***Comment*** |
| ConfirmTime | Integer | 500 | Time in milliseconds for a command to be confirmed |
| RespTime1 | Integer | 1000 | Response time in milliseconds |
| RespTime2 | Integer | 1000 | Response time in milliseconds |
| TempErr | Integer | 2 | Allowable error for "equal" temperatures |

| *Monitored Variable Dictionary* | | | |
|---|---|---|---|
| ***Name*** | ***Type*** | ***Initial Value*** | ***Comment*** |
| mAck | CommandType | None | Confirmation of change of fan speed by Incubator |
| mCancelFan | Boolean | False | Indicates setting fan speed has been cancelled |
| mCancelSciTemp | Boolean | False | Indicates setting science temperature has been cancelled |
| mCommand | CommandType | None | User command to the Incubator |
| mDoor | DoorType | Closed | Incubator door status |
| mEnableFan | Boolean | False | With toggle, fan speed change is enabled |
| mEnableSciTemp | Boolean | False | With toggle, science temperature change is enabled |
| mFanSpeed | Fan Speed | Off | Chamber fan speed |
| mRefFanSpeed | Fan Speed | Off | Reference fan speed desired for new chamber fan speed |
| mRefSciTemp | Temperature | 20 | Reference science temperature desired for new chamber science temperature |
| mSciTemp | Temperature | 20 | Average chamber science temperature |
| mTimed | Boolean | False | Indicates if command being sent is timed (true) or un-timed (false) |
| Time | Integer | 0 | Clock time in milliseconds |

| Controlled Variable Dictionary | | | |
|---|---|---|---|
| ***Name*** | ***Type*** | ***Initial Value*** | ***Comment*** |
| cAck | CommandType | None | Display of fan speed change confirmation |
| cCommand | CommandType | None | Command to the Incubator relayed by the Incubator UPA |
| cDoor | DoorType | Closed | Display of Incubator door status |
| cFanSpeedCommand | FanSpeed | Off | Display of chamber fan speed |
| cRefFanSpeed | FanSpeed | Off | Reference fan speed sent to the Incubator along with Command=SetFan |
| cRefSciTemp | Temperature | 20 | Reference science temperature sent to the Incubator along with Command = SetScienceTemp |
| cSciTemp | Temperature | 20 | Display of average chamber science temperature |
| cTimed | Boolean | False | Indicates if command being sent is timed (true) or un-timed (False) |
| cWarning | Boolean | False | Warning conditions |

| *Assumption Dictionary* | | |
|---|---|---|
| ***Name*** | ***Expression*** | ***Comment*** |
| ConfirmDeadlineFan | Duration (cCommand=setFan)>= ConfirmTime=>mAck=SetFan | Confirmation of command to set fan speed occurs no more than ConfirmTime after the command is sent |
| ConfirmDeadlineTemp | Duration(Command=SetRefTemp) >=ConfirmTime=> mAck=SetRefTemp | Confirmation of command to set reference science temperature occurs no more than ConfirmTime after the command is sent |

The tables below illustrate the input parameters for each of the modes. Detailed "views" of the results are presented in the *Incubator Display Software Cost Reduction Toolset Software Requirements Specification,* Appendix C, SCR Toolset Simulation Interface.

| *Event Function for cCommand* | |
|---|---|
| ***Type Command*** | ***Event*** |
| cCommand'=mCommand' | @C(mcExecution Status) OR @(mcTimedExecutionStatus) |

| *Mode Transition Function for mcExecutionStatus* | | |
|---|---|---|
| ***Source Mode(s)*** | ***Events*** | ***Destination Mode(s)*** |
| Timed | @T(mCommand!=None) | Immediate1 |
| Immediate1 | @T(mCommand!=None) | Immediate2 |
| Immediate1 | @T(mCommand=None) | Timed |
| Immediate2 | @T(mCommand=None) | Immediate1 |

| *Mode Transition Function for mcFanMode* | | |
|---|---|---|
| ***Source Mode(s)*** | ***Events*** | ***Destination Mode(s)*** |
| Idle | @C(mEnableFan) | FanChangeEnabled |
| FanChangeEnabled | @T(mCommand=SetFan) WHEN (NOT mTimed) | CommandSent |
| FanChangeEnabled | @C(mCancelFan) | Idle |
| CommandSent | @T(mAck=SetFan) WHEN (mFanSpeed!= mRefFanSpeed) | ChangingFan |
| CommandSent | @T(mAck=SetFan) WHEN (mFanSpeed= mRefFanSpeed) | Idle |
| CommandSent | @C(time) WHEN (duration(Inmode) = RespTime1-1) | FanError |
| ChangingFan | @T(mFanSpeed = mRefFanSpeed) | Idle |
| ChangingFan | @C(time) WHEN (duration(Inmode) = RespTime2-1) | FanError |

Explanation of transitions, in descending order:

- The user has enabled a chamber fan speed change.
- @T(mCommand=SetFan) corresponds to pushing the "Send" button in the Send Command box on the GUI. The user has also chosen an immediate un-timed command.
- @C(mCancelFan) cancels the Send Command.
- The Incubator confirms that the command was sent. The Display is now waiting for the chamber fan speed to change.
- The Incubator confirms that the command was sent, but the current fan speed is already the desired fan speed.
- If the Incubator has not responded after response time 1 has passed, there is an error.
- When the chamber fan speed reaches the reference value the task is complete
- If the Incubator has not responded after response time 2 has passed, there is an error.

| *Mode Transition Function for mSciTempMode* | | |
|---|---|---|
| ***Source Mode(s)*** | ***Events*** | ***Destination Mode(s)*** |
| Idle | @C(mEnableSciTemp) | TempChangeEnabled |
| TempChangeEnabled | @T(mCommand=Set RefTemp) WHEN (NOT mTimed) | CommandSent |
| TempChangeEnabled | @C(mCancelSciTemp) | Idle |
| CommandSent | @T(mAck=SetRefTemp) WHEN (abs(mRefSciTemp – mSciTemp) > TempErr) | ChangingSciTemp |
| CommandSent | @T(mAck=SetRefTemp) WHEN (abs(mRefSciTemp - mSciTemp)<=TempErr) | Idle |
| CommandSent | @C(time) WHEN (duration(Inmode) = RespTime1-1) | SciTempError |
| ChangingSciTemp | @T(abs(mRefSciTemp-mSciTemp) <=TempErr) | Idle |
| ChangingSciTemp | @C(time) WHEN (duration(Inmode) = RespTime2-1) | SciTempError |

Explanation of transitions, in descending order:

- The user has enabled a science temperature change.
- @T(mCommand=SetRefTemp) corresponds to pushing the "Send" button in the Send Command box on the GUI. The user has also chosen an immediate un-timed command.
- @C(mCancelSciTemp) cancels the Send Command.
- The Incubator confirms that the command was sent. The Display is now waiting for the science temperature to change.
- The Incubator confirms that the command was sent, but the current science temperature is already within TempErr of the reference science temperature.
- If the Incubator does not respond after response time 1, there is an error.
- When the chamber temperature reaches the new reference temperature (within a tolerance given by TempErr), the task is complete.

- If the Incubator cannot change the chamber temperature to the new reference temperature after response time 2 has passed, there is an error.

| *Mode Transition Function for mcTimedExecution Status* | | |
|---|---|---|
| ***Source Mode(s)*** | ***Events*** | ***Destination Mode(s)*** |
| Stopped | @T(mCommand=StartExp) | Running |
| Running | @T(mCommand=SuspendExp) | Suspended |
| Running | @T(mCommand=StopExp) | Stopped |
| Suspended | @T(mCommand=ResumeExp) | Running |
| Suspended | @T(mCommand=StopExp) | Stopped |

# APPENDIX C: SCR TOOLSET SIMULATION INTERFACE

Note: The SCR Toolset Simulation Interface is provided as a separate pdf document.

# Building a Graphical User Interface (GUI) for the SCR Simulator

## 1. Overview

This document presents a tutorial introduction to the graphical interface building features of the Java-based SCR Toolset. The user is provided with a palette of buttons, sliders, and other widgets that may be superimposed over graphics backgrounds to create a graphical interface to the SCR Simulator for an SCR model. Straightforward property boxes associated with each button may then be filled in to link widget actions to monitored and controlled variables of the SCR specification.

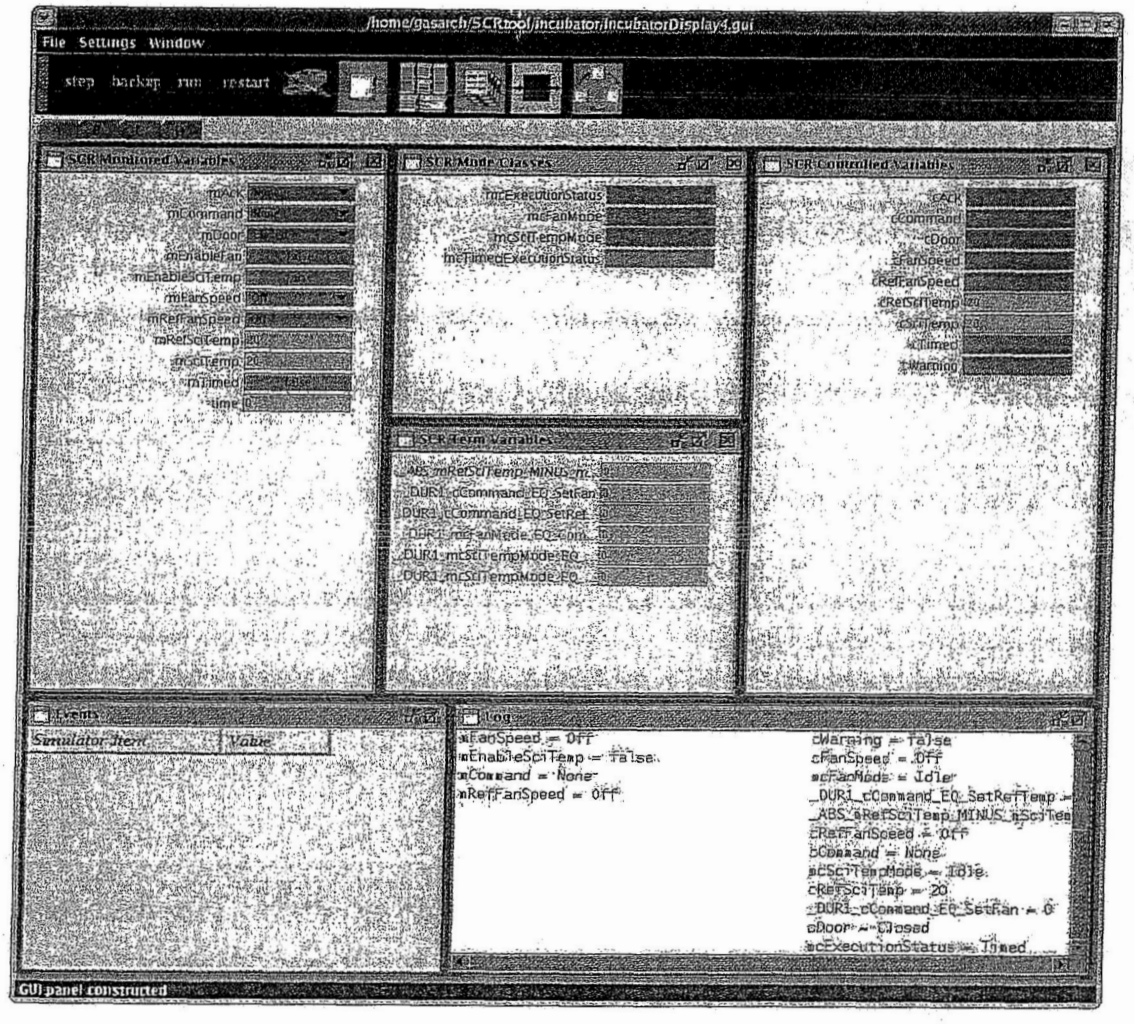
The tutorial leads the user through all of the steps in creating a GUI for the Incubator Display specification. Dumps of all relevant screens and property boxes are provided as the user follows the steps for the creation of the GUI. At any time during the process the user may exercise completed widgets to verify that they are properly connected to the Simulator.

## 2. The Simulator Interface

Upon starting the SCR simulator a default layout is presented as shown in the figure below. The default window lists all of the monitored, controlled, and term variables, as well as the mode classes and their values in a convenient format. There is also an **Events** window which lists all the events that have been run or queued up so far, and a **Log** window which shows the values of all variables which have changed during each step.

Just above the the simulation window are four tabs labeled **A**, **B**, **C**, and **D**. There are no distinctions between the tabs (they may be renamed) EXCEPT that the **Log** and **Events** windows may only appear in the first tab, by default called tab **A**. This distinction will become important later when building the GUI.

Along the top of the of the window a toolbar is displayed. The buttons **Step**, **Backup**, **Run**, and **Restart** control the progress of the simulation. The next button to the right, which looks like a paint palette, will open up a palette of widgets which can be used to create new GUI objects. The next button to the

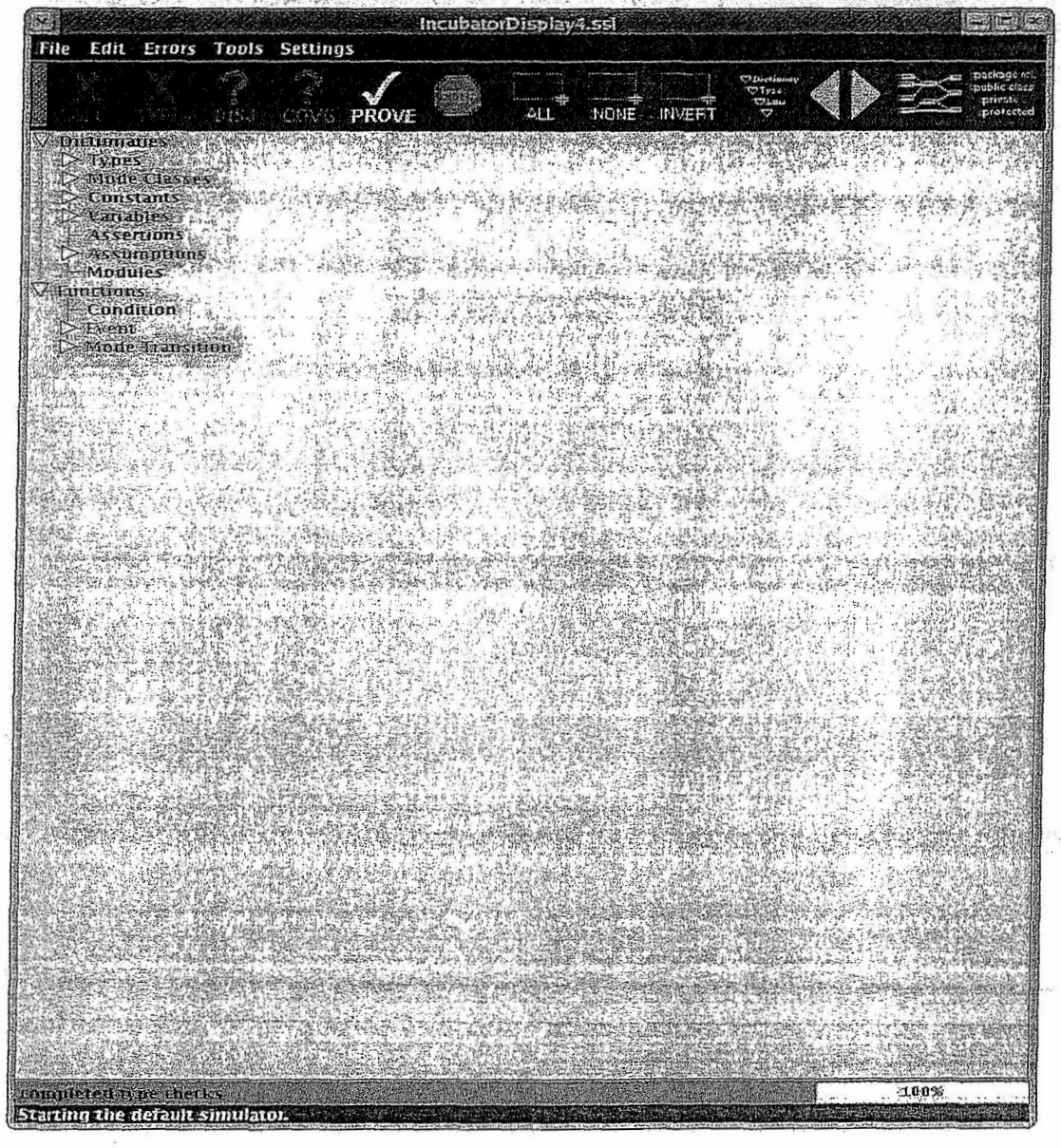



right is the create new canvas button which will create a new blank canvas in the current active tab. The next button to the right will recreate the default view in the currently active tab. This is very useful function when switching between the GUI and default views. The remaining three buttons on the toolbar are under development and should not be used.

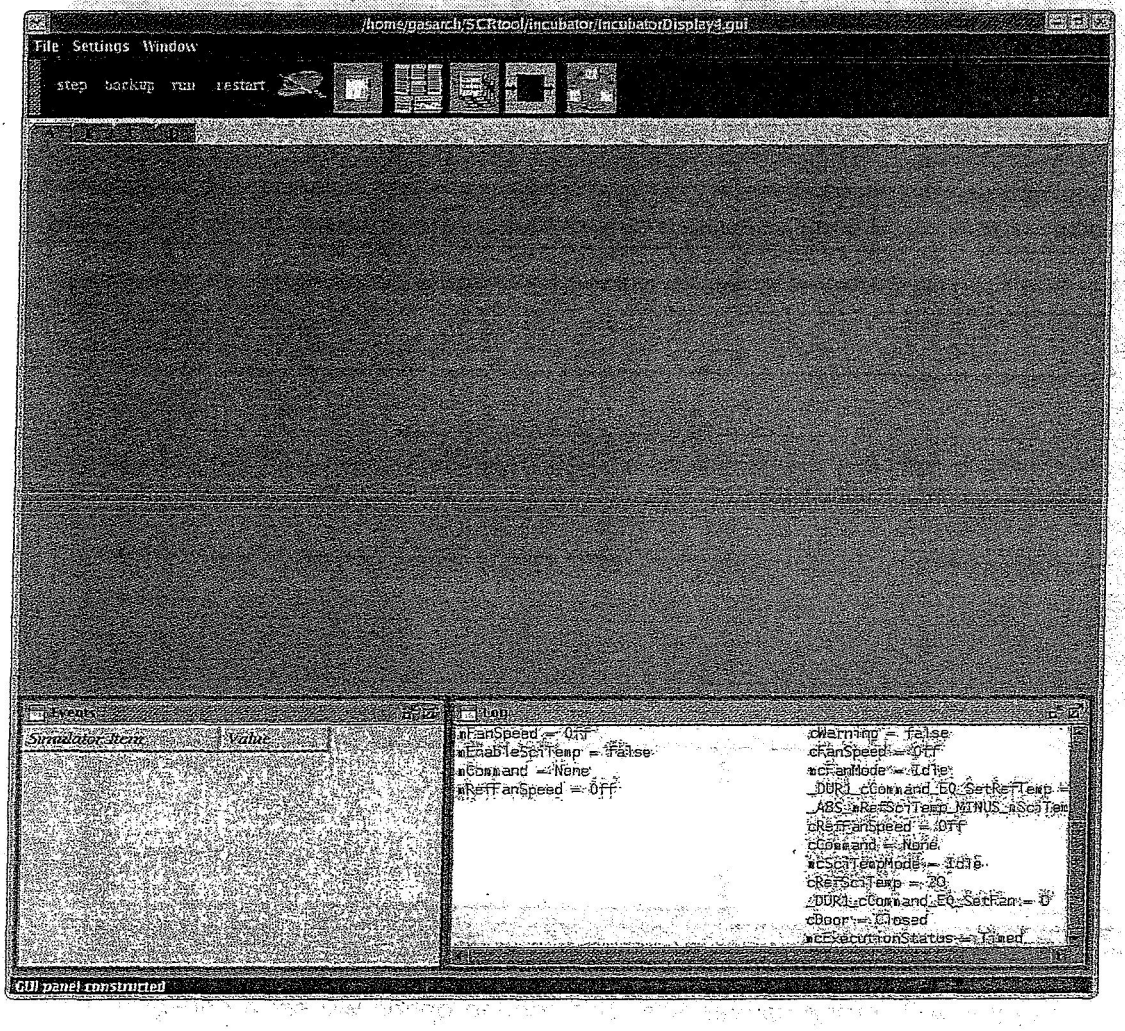
## 3. A Tutorial Introduction

This tutorial will show how to create a GUI for the `IncubatorDisplay4.ssl` specification example. GUI building is a mostly point and click process and once finished with this tutorial introduction it should be easy to complete the rest of the windows in a similar fashion.

Start off by opening the `IncubatorDisplay4.ssl` specification file and clicking on the simulator icon on the far right of the toolbar ("package nrl").

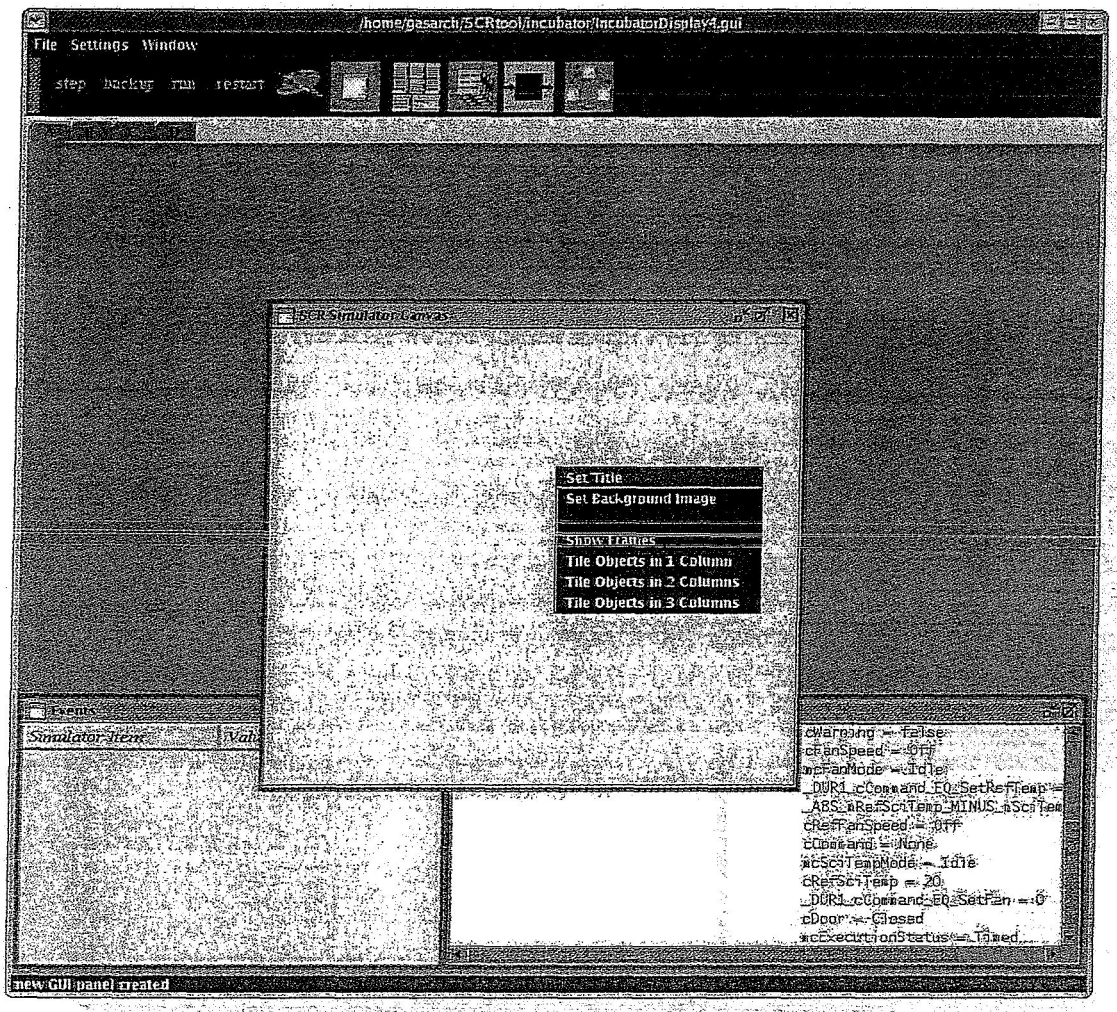



Now this is the part where the tabs become important. Since the **Log** and **Events** windows can only appear on the first tab it is standard practice to build the GUI on the first tab. The reasoning behind this is that when demoing a specification one generally would show the GUI and so it would be nice to see the **Log** and **Events** windows on the same pane as the GUI. Close all the windows except for the **Events** and **Log** windows.




Now create a blank canvas by clicking on the canvas button. Each tab may contain an unlimited number of canvases and each canvas may have its own background image and title.

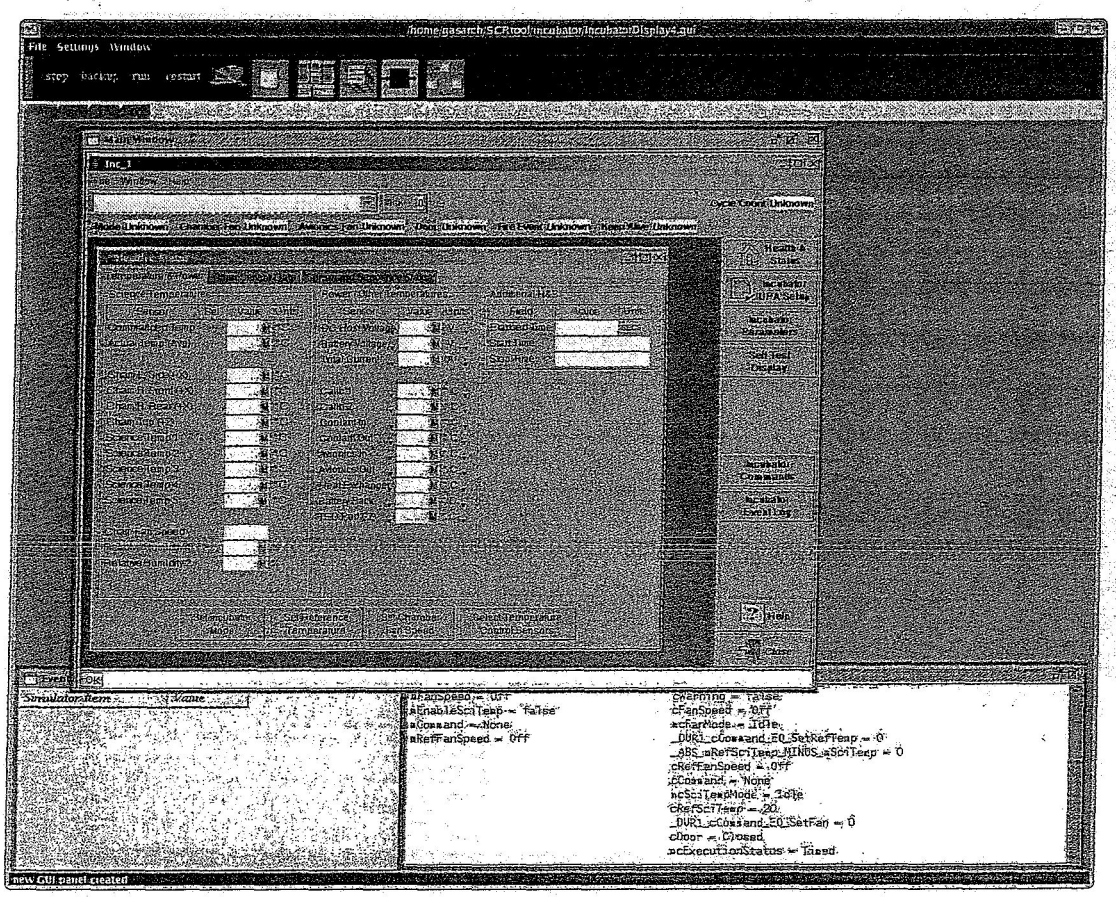
Now right click anywhere in the white space of the canvas. A pop up menu with seven options will appear.




The first option will set the canvas title. The second option will set a background image for the canvas. The third option will tell the program whether or not to tile the background image. The fourth option will show and hide frames for the widgets on the canvas.

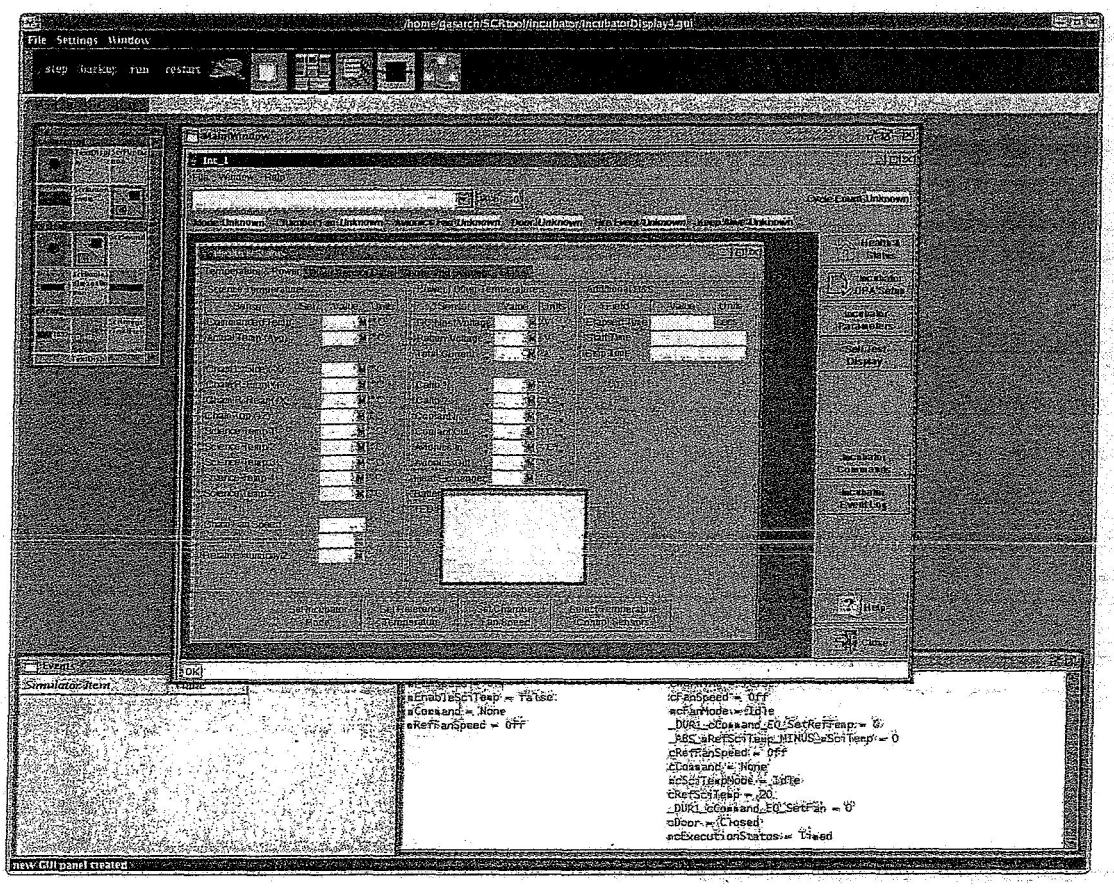
***WARNING:*** Generally, avoid the fifth, sixth, or seventh options. These will destroy all the placements and resizing options you have made to the widgets on the canvas without a dialog.

Now set the title to **Main Window** and set the background image to **`screen1.gif`**. Resize the canvas so that the entire background image is visible.



Now right click on the background again and click **Show Frames.** This will later cause all widgets to be surrounded by a frame, which facilitates moving and resizing.

Click on the palette button in the toolbar. The palette contains widgets for the four base type variables in SCR. Each widget is specific to its own type of variable. So for example, a boolean widget CANNOT be used for an integer base type variable.

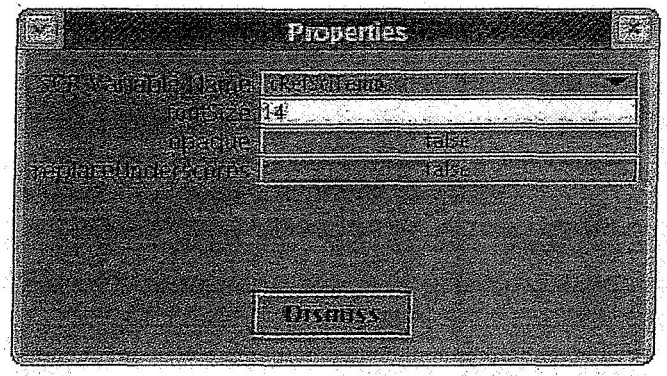


In the Integer widgets, click and drag onto the canvas the widget labeled TextDisplay. This is a simple widget that will display the value of a variable but not allow editing of the value.

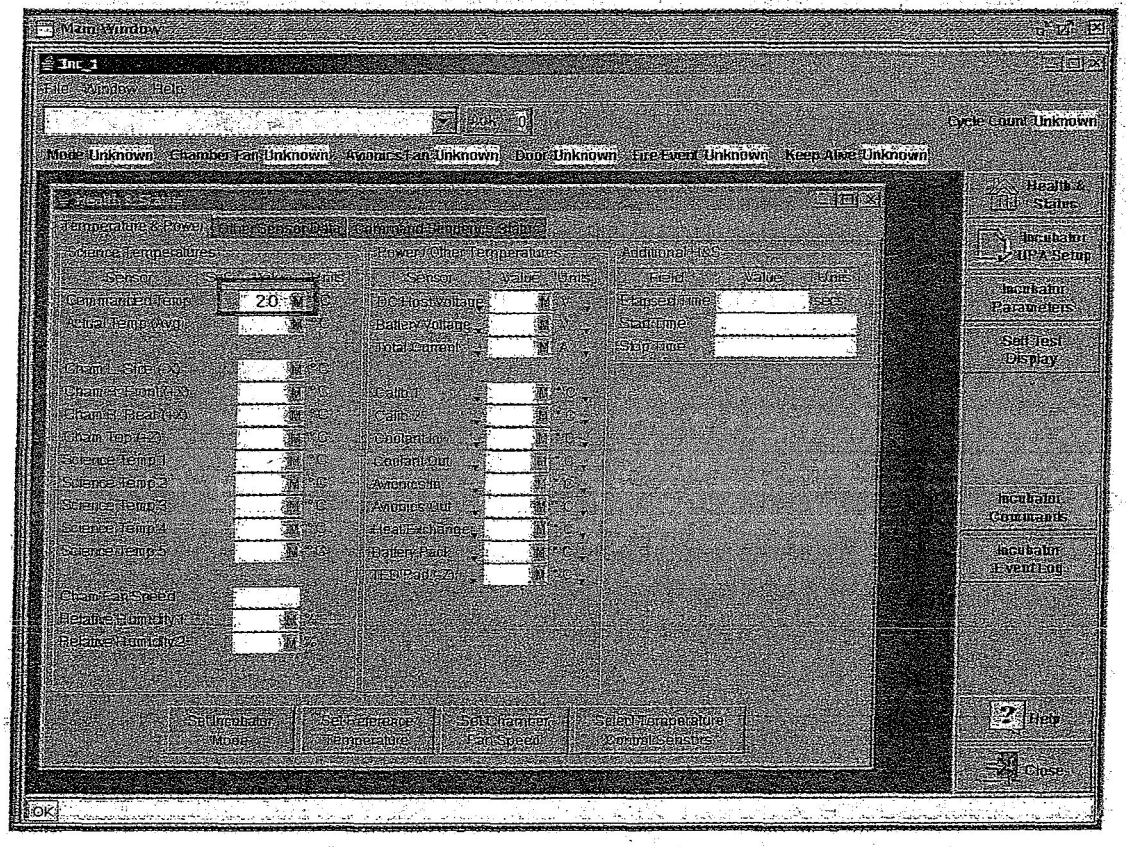
A white box with a dark gray frame around it should now be visible. If not right click anywhere on the canvas and click **Show Frames** to reset this feature.

Right click on the edge of a frame and a different pop up menu should appear with two options. Click on **Properties** and this will bring up a dialog box.

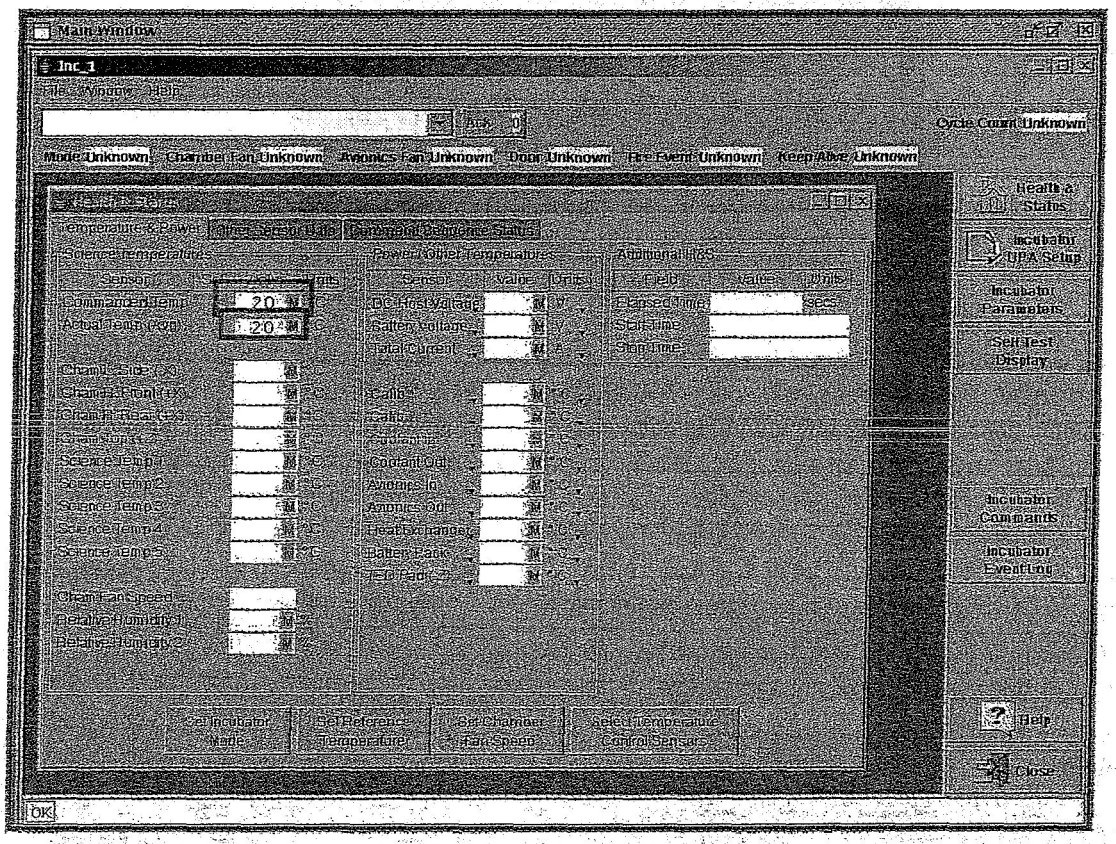
Each widget has its own different properties. In the case of a TextDisplay, the first property sets what variable this widget is associated with. The second property sets the font size for the widget's text. The third property sets whether or not the widget should be opaque. Set the SCR Variable Name to *cRefSciTemp*, font size to 14, and opaque to false.




Now the widget is ready to be placed into position. Resize the widget by placing the mouse cursor over one of the edges of the frames. When the cursor changes, then click and drag to change the size as desired. In order to move the widget place the mouse cursor just under the top edge of the frame. The cursor should change into a four-directional arrow. Now resize the TextDisplay to fit into the area just to the left of Command Temp on the canvas.



Do the same with Actual Temp except setting the variable to *cSciTemp*.




For enumerated types, the process is exactly the same, except remember that an enumerated type widget is required. Use the equivalent TextDisplay for enumerated types to finish creating Cham Fan Speed, and Door values.

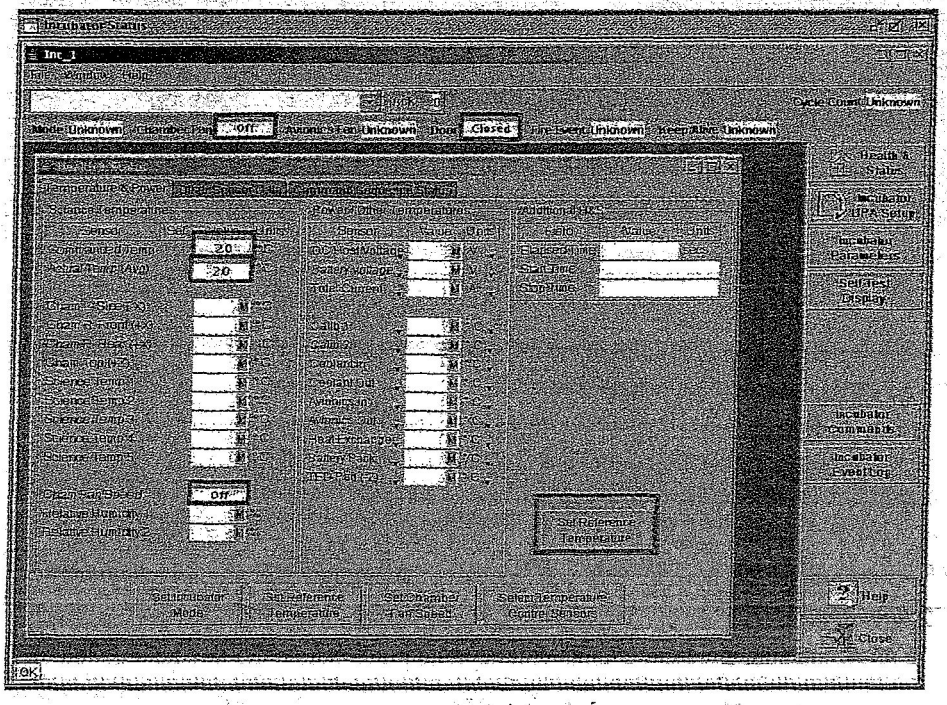
**TIP:** While not necessary, it is recommended to resize the widgets as small as possible or to follow the natural boundaries as close as possible so that a user running the demo won't accidentally move or resize the widget.

Widgets that only output information are pretty general and only one or two different ones are needed in most cases. A Widget that takes input, however, can be very specialized, depending on the desired behavior and the type of variable it affects, so care must be taken in choosing an appropriate widget.

For example, for the button **Set Reference Temperature**, the action of depressing the button should set *mEnableSciTemp* to TRUE. This should also simulate a button, so one way to do this is to use ScrBooleanImageButton and show two different images for the released and clicked state.

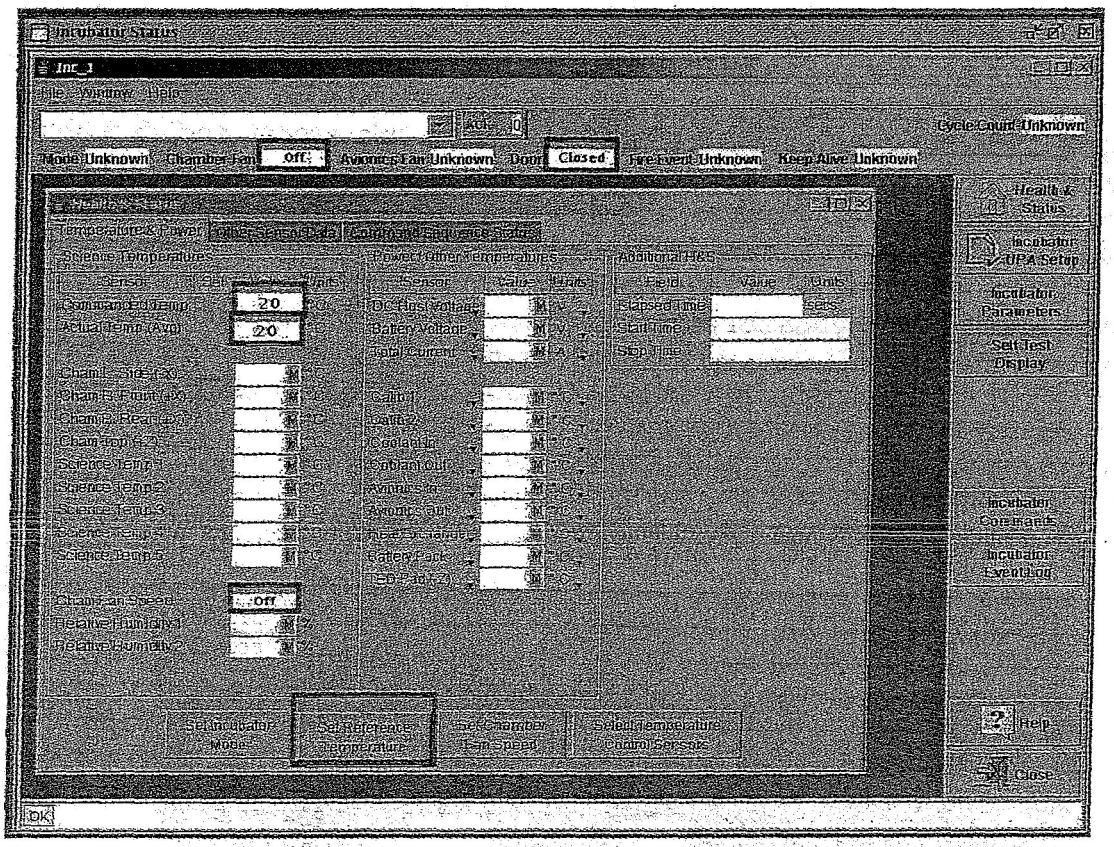
Create a new ScrBooleanImageButton and go to its properties. Choose *mEnableSciTemp* for the SCR variable name. If that name is not in the list, a widget of the wrong type was chosen, so delete this widget and find an appropriate one. For **imageDown** and **imageUp** type in the a path to the images to be displayed when the button is clicked or released. Note that this path is relative to the directory you started the original SCRtool in. For this example it is assumed that the SCRtool was started in its main directory, so set the path of **imageDown** to `incubator/setreftemppressed.gif` and set **imageUp** to `incubator/setreftemp.gif`[1]. The last property, **whatToReturn** is the value this widget should set the variable to when the button is clicked. In this case, *mEnableSciTemp* should be set to true when this button is clicked so set **whatToReturn** to true. Finally for aesthetics, set opaque to false.

The button obtained should look like the original in the background.



Resize and move the new button over the original.

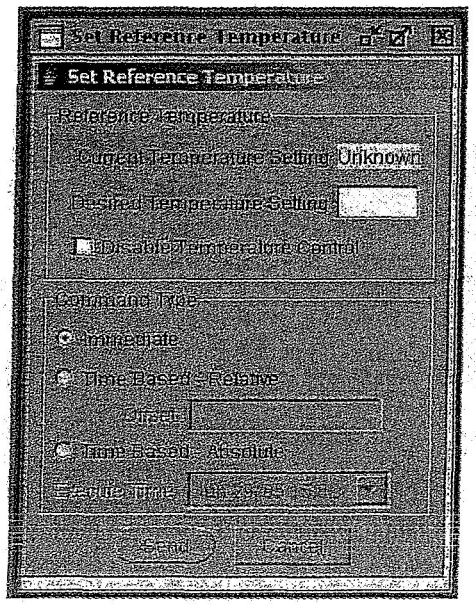
1 Use of this widget requires creation of the appropriate images by some image manipulation program such as GIMP. Such image creation is out of the scope of this tutorial.




Do the same with Set Chamber Fan Speed, only instead of changing *mEnabledSciTemp*, change *mEnableFan.*

This is the completion of the first window! Don't forget to save this GUI design. Test it by removing the frames and clicking on the buttons. Clicking a button that changes the value of any variable, will be indicated by green highlighting of that variable. The corresponding changes will also be displayed in the **Event** and **Log** windows. Now continue creating the rest of the windows by creating new canvases.

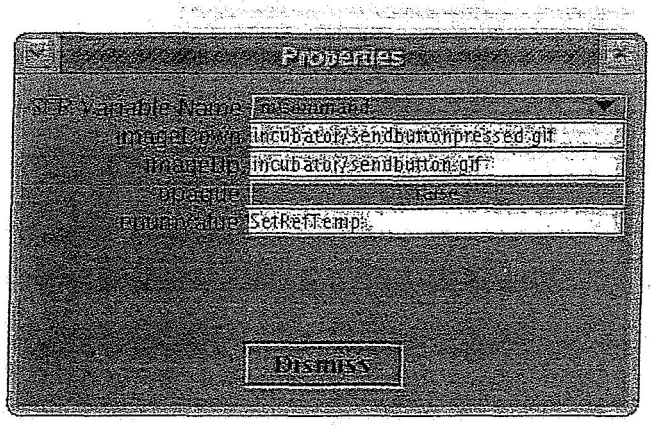
Next create a new canvas for the Set Reference Temperature dialog window. This canvas will have a total of four widgets, one to display the current temperature setting and three for input. Set the background image to `screen2.gif`.



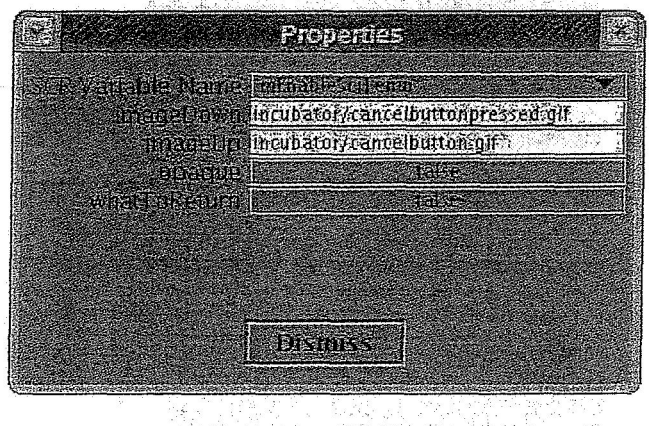

Now add a TextDisplay widget to the **Current Temperature Setting** space. The widget must not be transparent, otherwise the "Unknown" will show through from the background image. **Desired Temperature Setting** should be ScrIntegerTextField to allow input by typing in numbers directly. Set its name to <space> since the name is already showing to the left in the background image.

The **Send** button at the bottom is like **Set Reference Temperature** of the previous window, but instead of a boolean, **Send** sets a enumeration. Use the equivalent widget, the ScrButtonEnum. For this button, set:

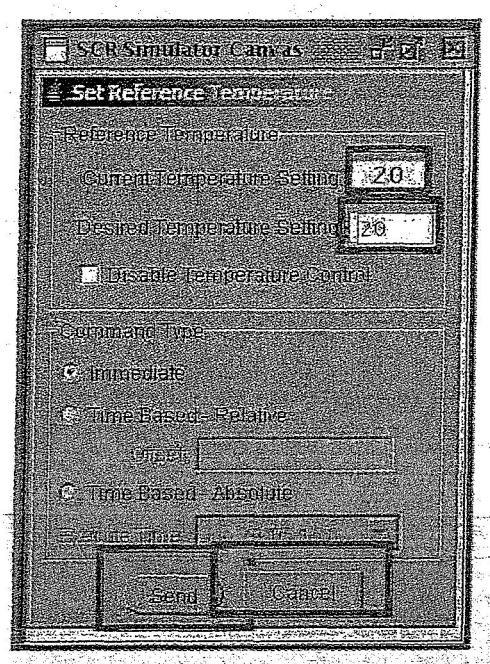
**SCR variable** to *mCommand*
**imageDown** to `incubator/sendbuttonpressed.gif`
**imageUp** to `incubator/sendbutton.gif`
**opaque** to false.
**enumValue** to SetRefTemp



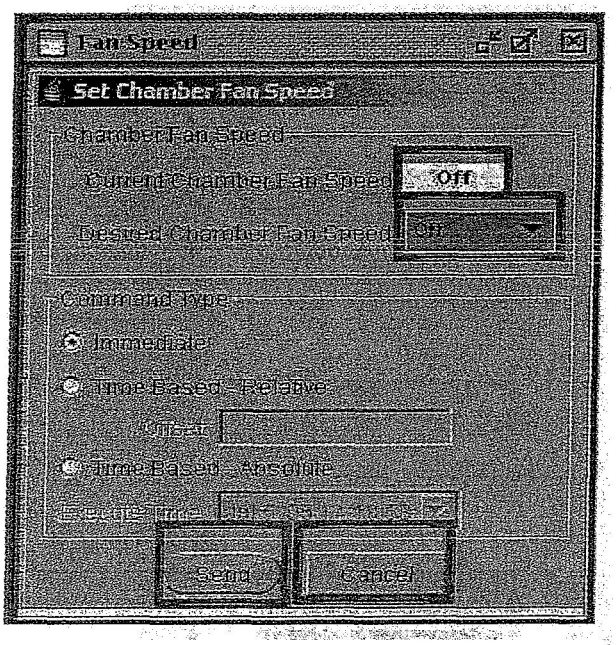
The **Cancel** button would also be similar, except it is a boolean variable that sets mEnableSciTemp to false.



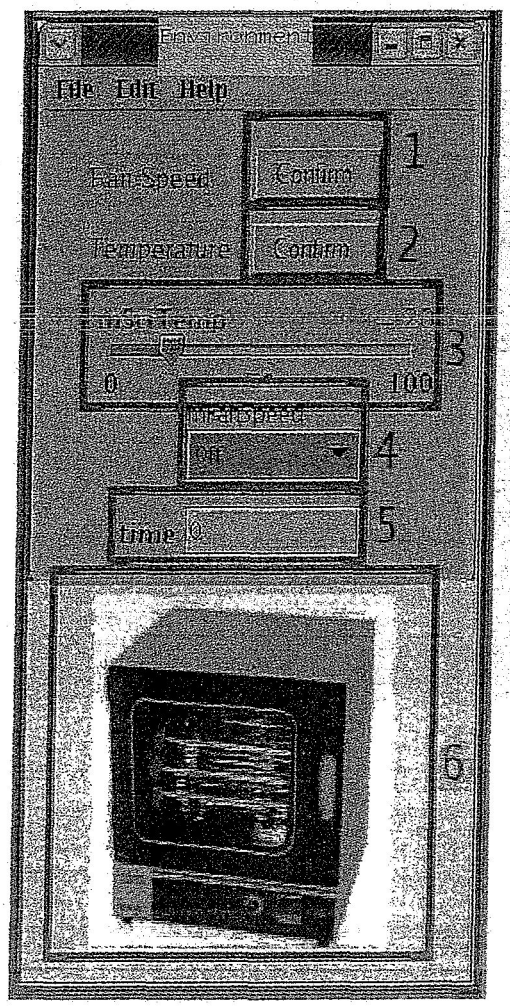

Finally the window should look like this when done:



For the Set Fan Speed window use the analogous widgets appropriate for enumerated types. The background image is located at `incubator/fanspeedbackground.gif`. The same images can be used for the **Send** and **Cancel** buttons. This is what the Fan Speed should look like when completed.

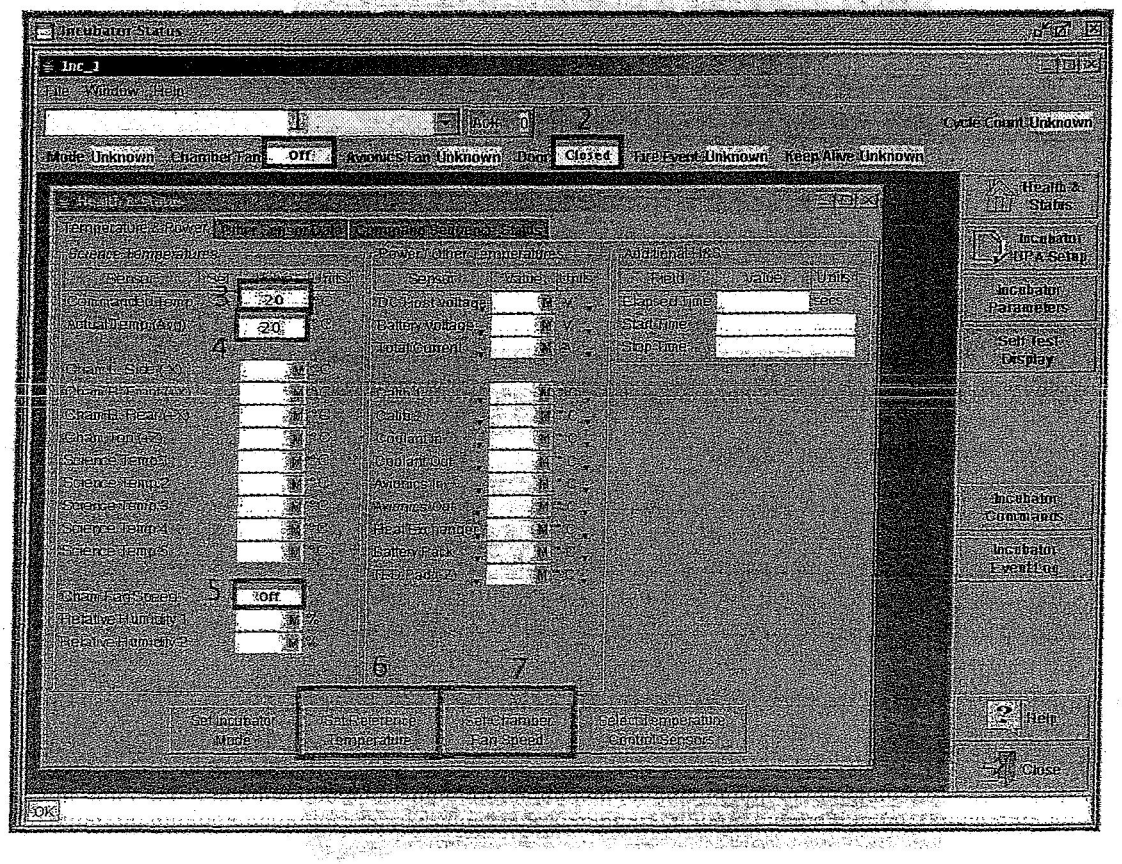


The environment window will have use six widgets. There are confirm buttons for both the Fan Speed and Temperature which are both ScrButtonEnums. The actual temperature setting uses an Integer Slider and the Fan speed is controlled via an Enumeration Combobox. The last item at the bottom is a ScrFlipButton for Enumerations.

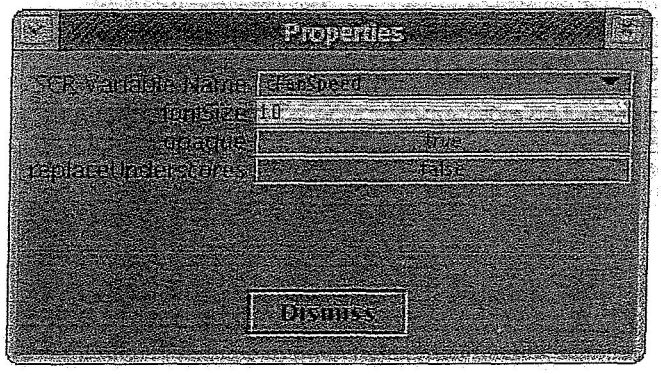


# Appendix A – Widgets and their appropriate settings.

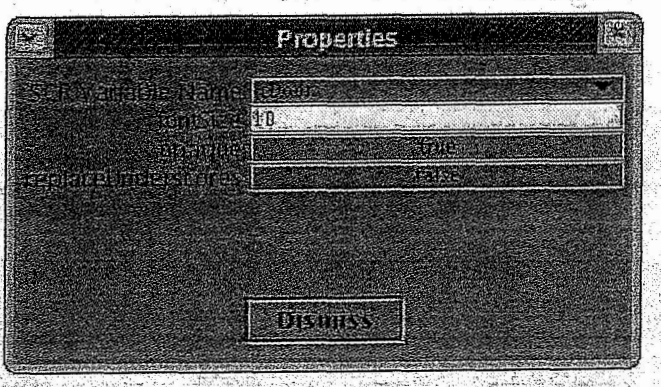
## Incubator Status Window



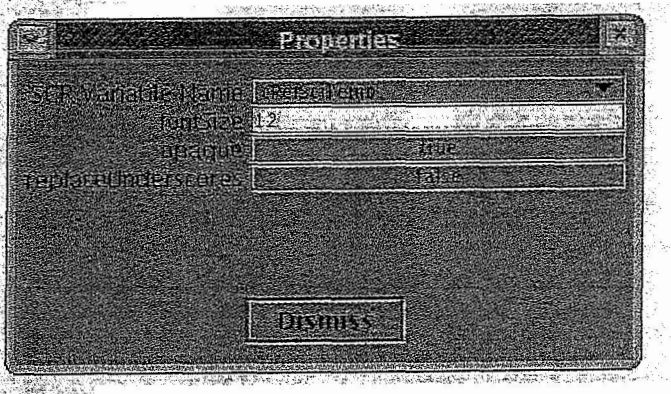

1. Enumerated TextDisplay



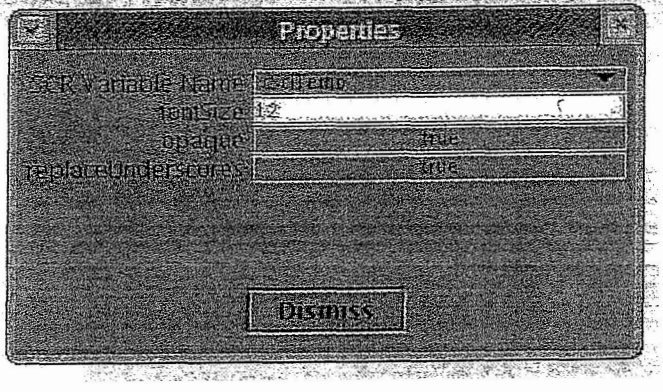
2. Enumerated TextDisplay



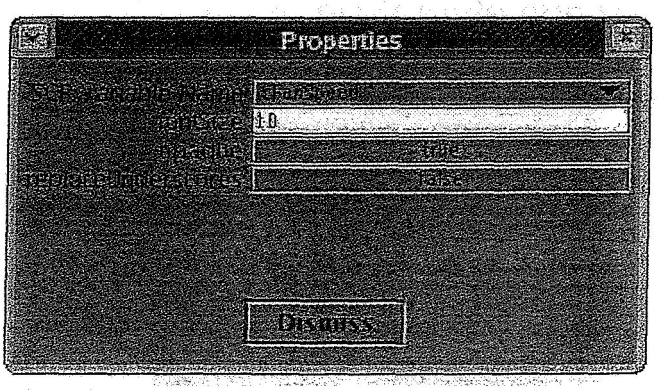

3. Integer TextDisplay



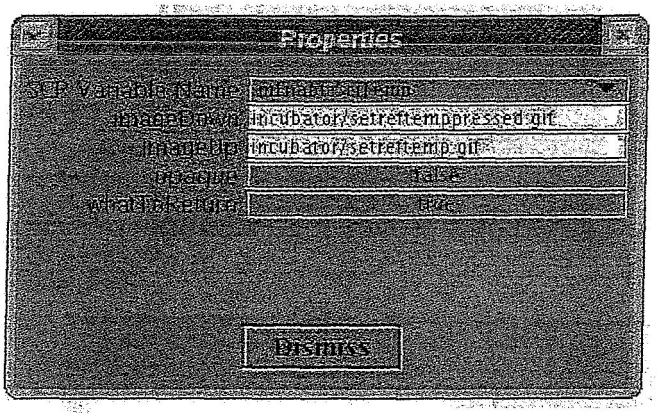

4. Integer TextDisplay



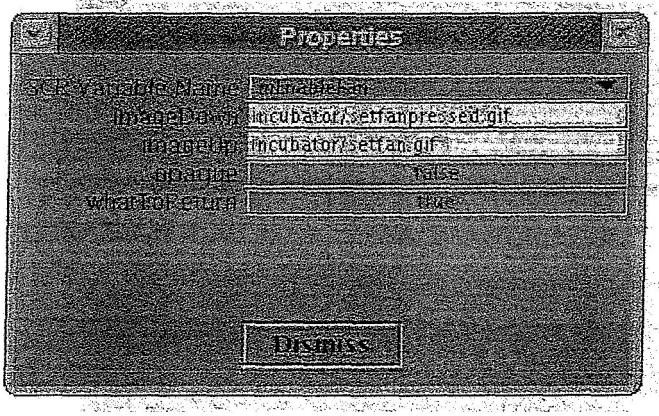
5. Enumerated TextDisplay



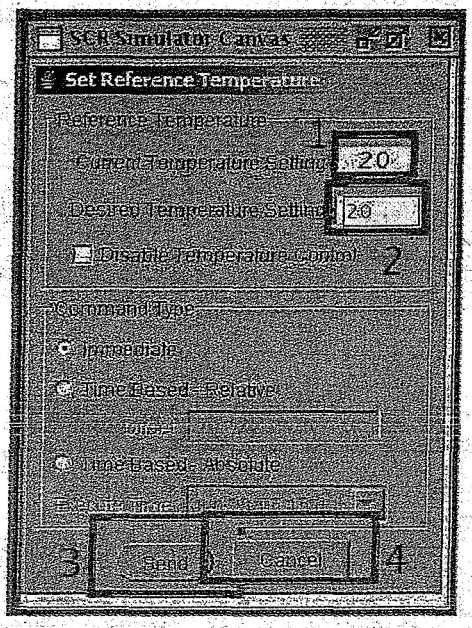

6. Boolean ImageButton



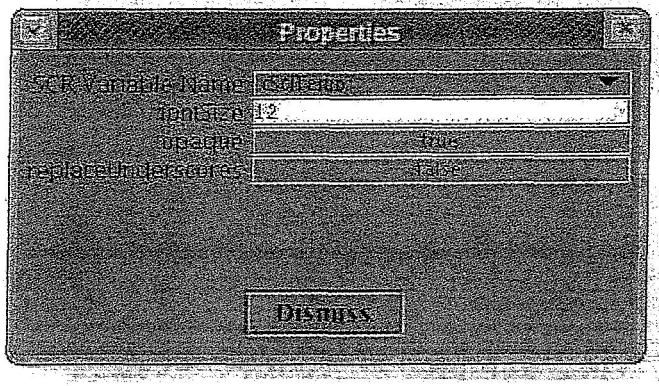

7. Boolean ImageButton



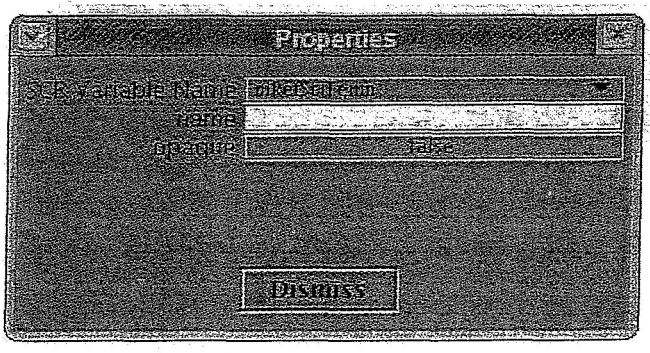
## Set Reference Temperature Window



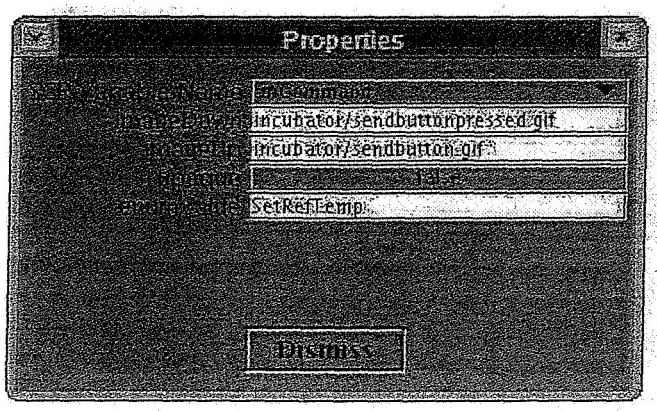

1. Integer TextDisplay




2. Integer TextField

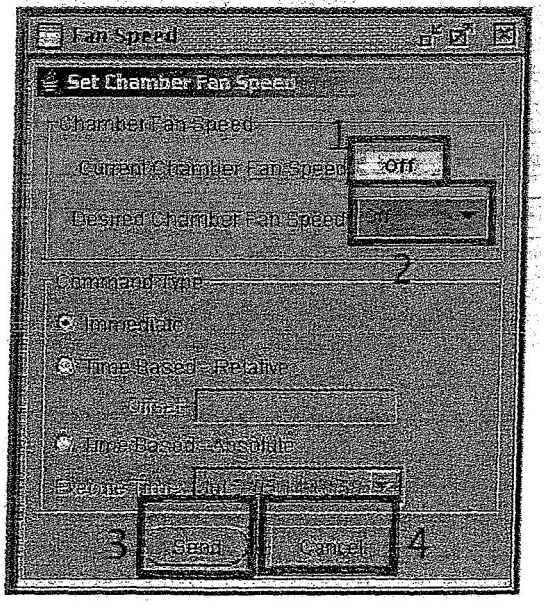


3. Enumerated ScrButtonEnum

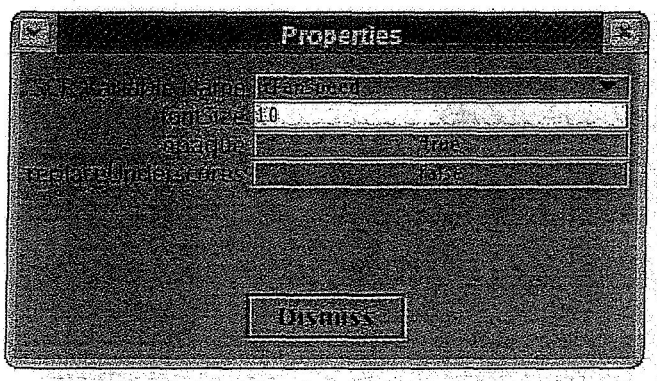



4. Boolean ImageButton

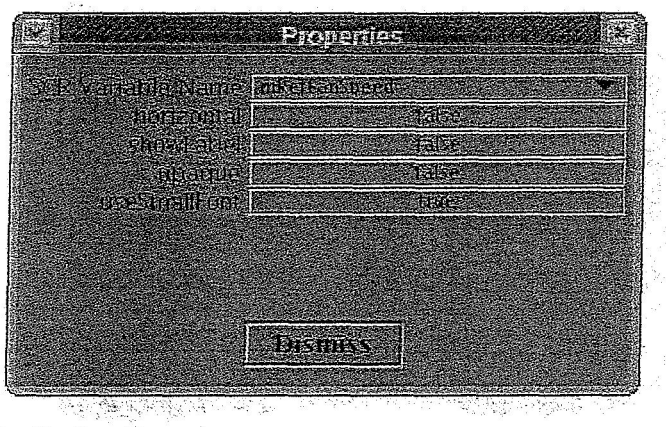

## Set Fan Speed



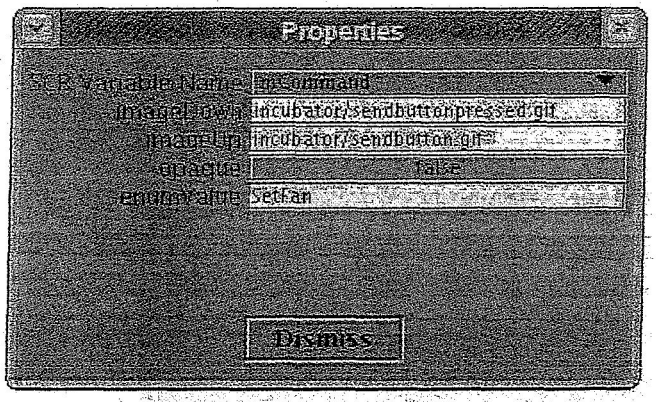
1. Enumerated TextDisplay



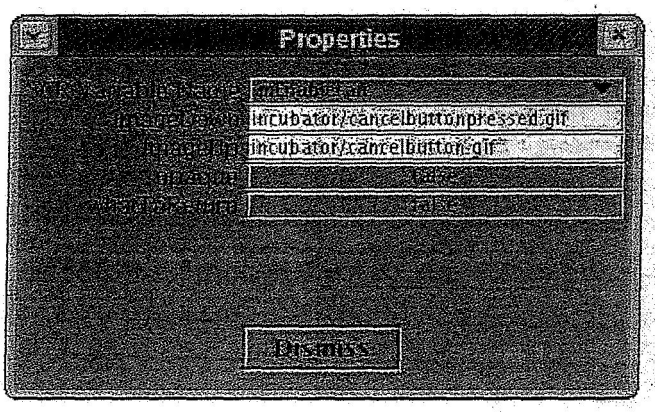

2. Enumerated Combobox



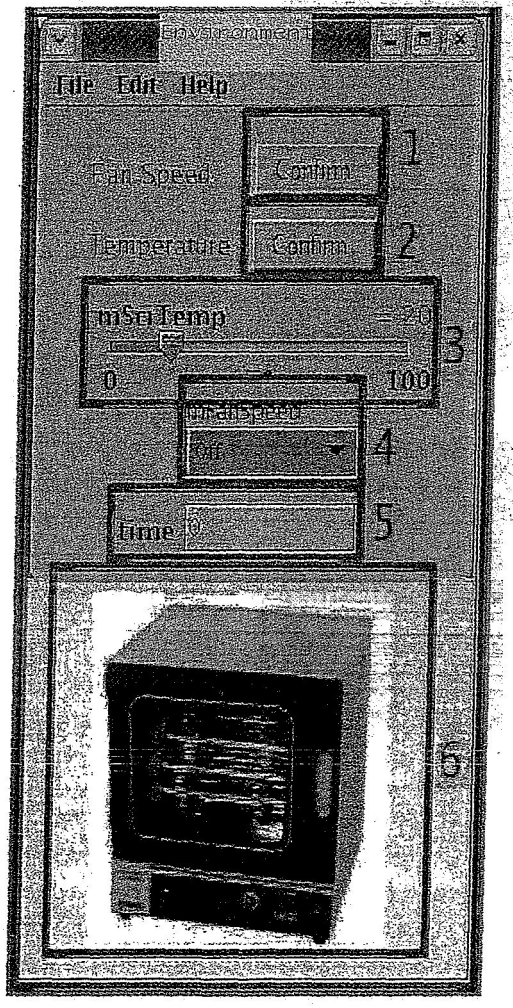

3. Enumerated ScrButtonEnum



4. Boolean ImageButton

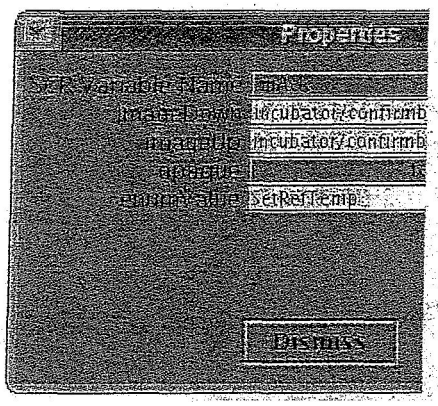



## Environment

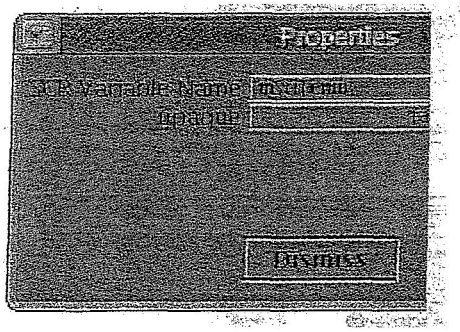


1. Enumerated ScrButtonEnum

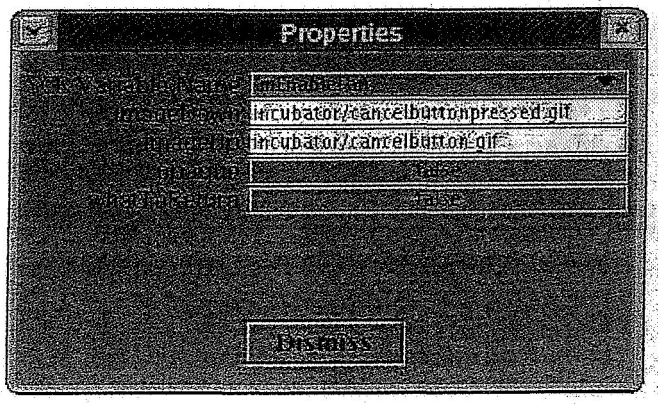

2. Enumerated ScrButtonEnum



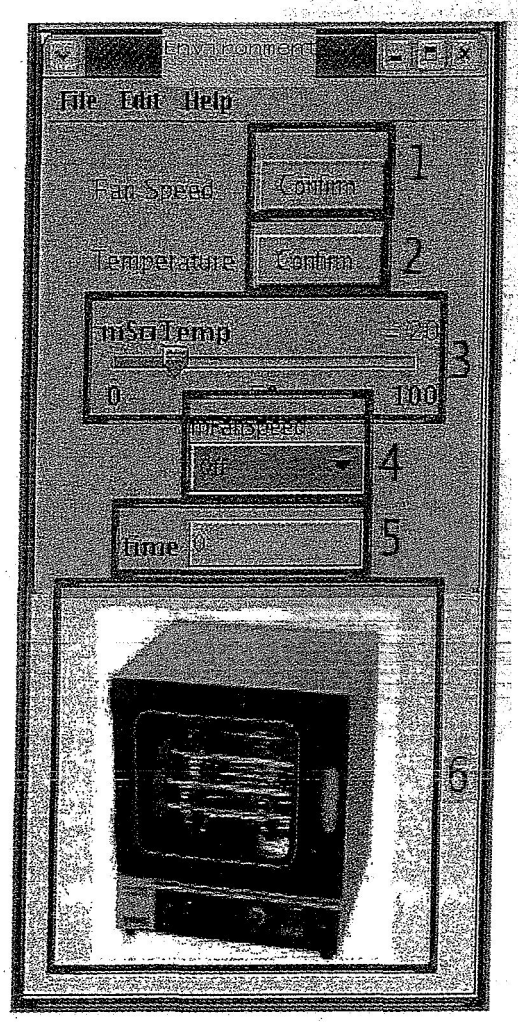

3. Integer Slider with Label



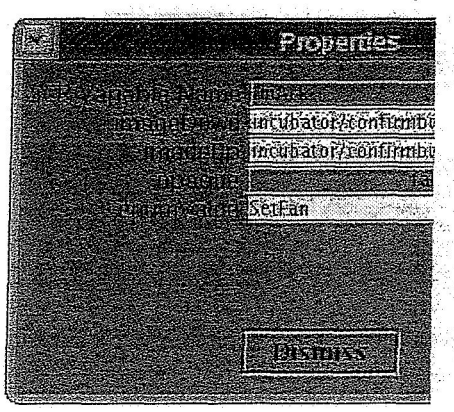
4. Boolean ImageButton



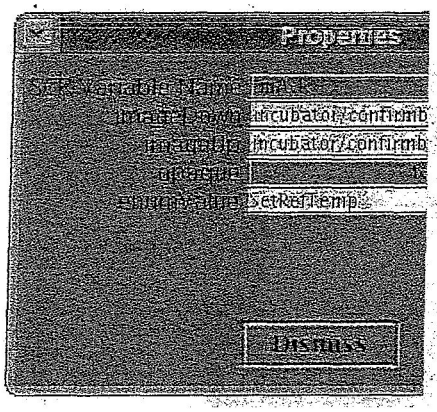

## Environment



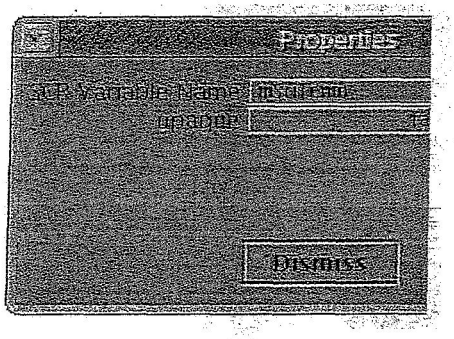
1. Enumerated ScrButtonEnum



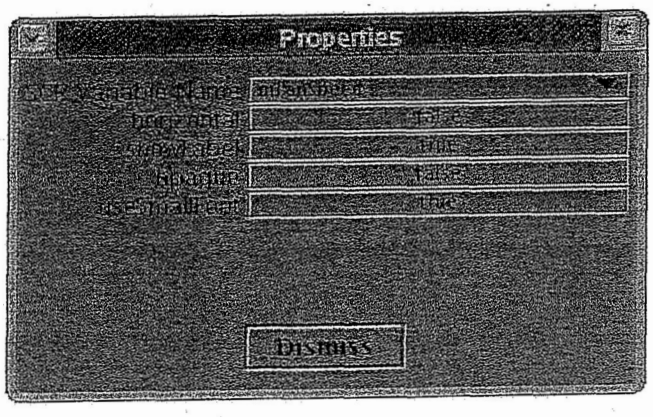

2. Enumerated ScrButtonEnum



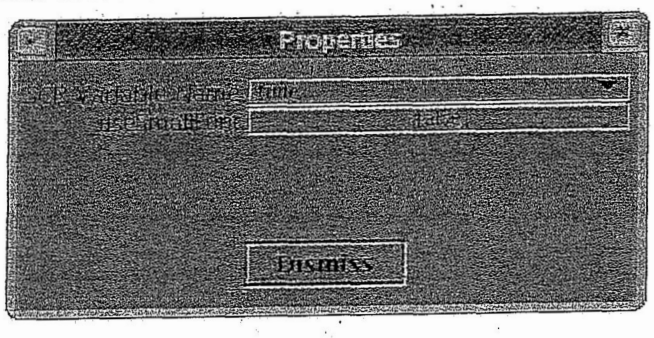

3. Integer Slider with Label



4. Enumerated Combobox




5. Integer TextField with Label




6. Enumerated FlipButton